Transfer learning for facial analysis with limited and inconsistent annotations

Martin Dornier

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

THÈSE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par
Martin DORNIER

Transfer Learning for Facial Analysis with Limited and Inconsistent Annotations

Thèse présentée et soutenue à Rennes, le 14 Décembre 2023
Unité de recherche : IRISA
Thèse N° : 23ISAR 38 / D23 - 38

Rapporteurs avant soutenance :

Véronique ÉGLIN Professeur des universités, INSA de Lyon
Laurent HEUTTE Professeur des universités, Université de Rouen Normandie

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Président :	Frédéric JURIE	Professeur des universités, Université de Caen Normandie
Examinateurs :	Véronique ÉGLIN	Professeur des universités, INSA Lyon
	Laurent HEUTTE	Professeur des universités, Université de Rouen Normandie
	Robert LAGANIÈRE	Professeur, Université d'Ottawa, Canada
	Christian RAYMOND	Maître de conférences, INSA Rennes
	Yann RICQUEBOURG	Maître de conférences, INSA Rennes
	Philippe-Henri GOSSELIN	Principal Scientist, InterDigital R&D France
Dir. de thèse :	Bertrand COÜASNON	Maître de conférences HDR, INSA Rennes

Remerciements

Je tiens d'abord à remercier tous les membres du jury. Je remercie Véronique Églin et Laurent Heutte qui ont accepté de rapporter cette thèse. Merci également à Frédéric Jurie et à Robert Laganière pour leur participation à la soutenance en tant d'examinateurs.

Mon directeur de thèse, Bertrand Coüasnon, mérite une reconnaissance particulière pour sa guidance experte et son soutien tout au long de ce parcours de recherche. Mes encadrants, Christian Raymond, Yann Ricquebourg et Philippe-Henri Gosselin, ont également joué un rôle essentiel en fournissant des conseils précieux et en partageant leur expertise.

Je remercie également l'ensemble des membres de l'équipe Shadoc. En particulier mes collègues doctorants avec qui j'ai pu partager le confort du bureau D271 : Killian Barrère pour sa jovialité (malgré quelques périodes dépressives sur la fin), William Mocaër toujours à la pointe de la technologie et Timothée Neithoffer pour ses performances lyriques.

Je n'oublie pas mes collègues d'InterDigital : Quentin Avril, François Leclerc, Glenn Kerbiriou et tant d'autres, que n'ai connus malheureusement que plus tardivement. Leurs connaissances, notamment dans le domaine des *computer graphics*, m'ont été fort utiles.

Enfin, je veux remercier mes amis et ma famille qui m'ont soutenu tout au long de cette aventure académique.

Cette thèse est le fruit d'un effort collectif et je suis honoré d'avoir eu l'opportunité de travailler avec des individus aussi remarquables.

Résumé en français

Les réseaux de neurones artificiels se sont rapidement développés dans de nombreux domaines ces dernières années, tels que la vision par ordinateur, le traitement du langage naturel ou le traitement audio. Cependant, la plupart des réseaux actuels sont entraînés de façon supervisée ce qui requiert des données annotées, particulièrement pour les gros réseaux qui ont beaucoup de paramètres à optimiser.

Obtenir des données annotées en larges quantités peut se révéler compliqué car le processus d'annotation est souvent manuel ce qui rend la tâche coûteuse. De plus, certaines annotations, notamment celles en lien avec des données 3D nécessitent un équipement coûteux tels que des scanners ou des dispositifs multicaméras. Cela implique également un environnement de capture contrôlé ce qui limite le nombre de sujets et donc la variété du jeu de données. Le manque de données annotées restreint le développement de l'apprentissage profond dans de nombreux domaines. Pour pallier ce problème, les chercheurs en apprentissage profond s'intéressent de plus en plus à l'apprentissage avec peu ou même sans données annotées. L'avantage des données non annotées est qu'elles peuvent être récupérées en grandes quantités, notamment sur Internet, et donc qu'il est facile d'obtenir des jeux de données d'entraînement très variés. Récemment, les modèles autosupervisés qui peuvent s'entraîner sur des données non annotées ont eu beaucoup de succès. Ces modèles tirent parti de leur large jeu de données d'entraînement pour apprendre des représentations compactes des données. Ils peuvent ensuite être adaptés à une tâche supervisée en utilisant peu de données annotées. Ce processus est appelé apprentissage par transfert car il y a un transfert des connaissances apprises par le réseau durant l'apprentissage autosupervisé avec la tâche supervisée.

Parmi les méthodes autosupervisées ayant émergé ces dernières années, la plupart d'entre elles réalisent un apprentissage par transfert à partir de représentations de petite dimension apprises lors de l'entraînement sur des données non annotées [Che+20a; Car+20]. Généralement, après cette première étape d'entraînement, un petit réseau auxiliaire, utilisant ces représentations apprises comme entrée, est entraîné de manière supervisée sur la tâche finale. Cette approche convient bien aux tâches où la prédiction du réseau auxiliaire est également de petite dimension, comme la classification d'images, mais

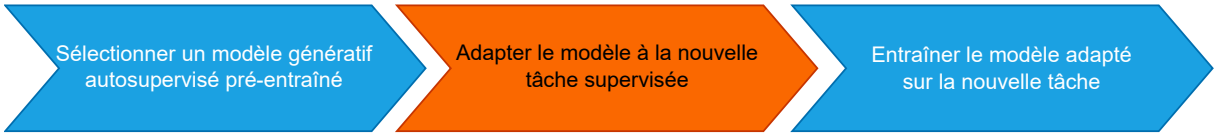


Figure A – Notre méthodologie GMDA pour entraîner un réseau avec peu de données annotées.

elle est moins efficace lorsque cette prédiction est de haute dimension, par exemple une image, car le réseau auxiliaire à entraîner est beaucoup plus grand. Celui nécessite donc un plus grand nombre d'échantillons annotés d'entraînement pour obtenir des performances convenables.

Cette thèse CIFRE, en collaboration avec l'entreprise InterDigital, a pour objectif de résoudre le problème de l'entraînement des réseaux de neurones, avec des données annotées limitées, pour des domaines d'intérêt d'InterDigital tels l'analyse faciale. En effet, de nombreuses tâches de ce domaine souffrent d'un manque d'annotations ce qui limite le développement de méthodes basées sur l'apprentissage profond pour ces tâches.

Nous proposons dans ce manuscrit une nouvelle approche permettant d'entraîner des réseaux de neurones avec des données annotées limitées pour certaines tâches type image-vers-image (la prédiction du réseau est une image). Nous suggérons d'utiliser non seulement des représentations de petite dimension, mais aussi des caractéristiques de grande dimension, issues du décodeur de modèles génératifs autosupervisés, lors de l'apprentissage par transfert. Nous avons appelé cette méthodologie l'Adaptation de Décodeur de Modèle Génératif (*Generative Model Decoder Adaptation* en anglais, GMDA). Nous démontrons également comment les prédictions et vérités terrain de certaines tâches d'analyse faciale supervisées peuvent être transformées en images, ce qui rend notre méthodologie applicable à ces tâches.

Notre méthodologie GMDA se décompose en plusieurs étapes.

— Choisir un réseau génératif autosupervisé pré-entraîné.
— Adapter l'architecture du réseau à la nouvelle tâche supervisée image-vers-image.
— Entraîner ce réseau sur la tâche image-vers-image en utilisant peu de données annotées.

Ces différentes étapes sont résumées dans la Figure A.

Dans un premier temps, nous avons appliqué notre méthodologie à la détection de points d'intérêt faciaux. Cette tâche consiste à prédire la position de points d'intérêt du visage tels que la position des yeux, du bout du nez ou des coins de la bouche. Annoter une

image pour la détection de points d'intérêt faciaux est fastidieux ce qui a pour conséquence que les jeux de données pour cette tâche sont relativement petits et donc les modèles appris dessus sont sujet au sur-apprentissage. Avec notre méthodologie, nous avons tenté de contourner ce problème du manque d'annotations.

Pour le réseau génératif, nous avons testé deux architectures, la version GMDA-R qui utilise un autoencodeur basé sur ResNet [He+16] proposé par Browatzki et al. [BW20] et la version GMDA-S qui utilise un autoencodeur basé sur StyleGAN [KLA19]. Le réseau StyleGAN possède une plus grosse capacité et est donc possiblement capable de modéliser des tâches plus complexes. Cependant il possède plus de paramètres à optimiser si un affinage (*fine-tuning* en anglais) est nécessaire lors de l'apprentissage supervisé, ce qui augmente le risque de sur-apprentissage si le nombre de données lors de cet apprentissage est trop faible.

Pour l'adaptation du réseau génératif à la tâche image-vers-image supervisée, nous utilisons les Couches de Transfert Entrelacées (*Interleaved Transfer Layers* en anglais, ITL) proposées par Browatzki et al. [BW20] dans leur architecture 3FabRec. Les ITL sont des couches de convolution ajoutées entre les couches du décodeur du réseau génératif et qui sont entraînées lors de l'apprentissage supervisé. Elles permettent de réutiliser les activations des couches du décodeur tout en les adaptant à la tâche supervisée. Seules les ITL ont besoin d'être entraînées à partir de zéro (l'encodeur peut éventuellement être affiné) ce qui permet d'entraîner le réseau avec peu de données annotées.

Nous proposons également des versions améliorées de ces ITL en rajoutant un flux direct entre les différentes ITL. De plus, nous suggérons d'améliorer les réseaux génératifs en rajoutant des connexions directes (*skip-connections* en anglais) entre les couches de l'encodeur et du décodeur.

Nous avons entraîné nos différents modèles sur différents jeux de données et avec différents nombres de données annotées d'entraînement. Puis nous avons comparé leurs performances entre eux ainsi qu'avec les méthodes de l'état de l'art existantes, en particulier celles qui s'entraînent avec peu de données annotées. Nous avons aussi testé l'utilisation de l'apprentissage actif pendant l'entraînement afin de réduire encore plus le nombre de données annotées nécessaire. Pour ce faire, nous proposons une nouvelle fonction d'acquisition, la Magnitude Négative de Voisinage (*Negative Neighborhood Magnitude* en anglais, NNM), pour évaluer les prédictions du réseau lors de l'apprentissage actif. La version GMDA-S obtient globalement des meilleurs résultats sauf quand le nombre de données annotées d'entraînement est très limité et les images de visage sont difficiles

(occultations, basse résolution...). Nos versions améliorées des ITL améliorent légèrement les résultats pour l'architecture GMDA-R. L'ajout des connexions directes bénéficie également à l'architecture GMDA-R. Enfin, l'usage de l'apprentissage actif permet d'améliorer les performances, en particulier pour les images difficiles où il permet sur certains jeux de données de diviser par deux le nombre de données annotées d'entraînement tout en obtenant la même performance en test. Comparés aux méthodes de l'état de l'art, nos modèles les surpassent sur de nombreux jeux de données quand le nombre de données d'apprentissage est limité.

Dans un second temps, nous avons testé notre méthodologie pour améliorer les méthodes autosupervisées de reconstruction faciale 3D. Cette tâche consiste à prédire la structure 3D d'un visage à partir d'une image. Obtenir des annotations de qualité pour cette tâche nécessite d'utiliser un scanner ce qui implique un environnement contrôlé et donc un nombre restreint de sujets. Pour contourner ce problème, des méthodes autosupervisées ont été développées ces dernières années. Cependant, comme leur fonction de coût est principalement basée sur la reconstruction de l'image, et qu'elles n'ont accès à aucune information 3D, elles ont tendance à prédire une mauvaise pose et échelle pour le visage 3D.

Pour aider ces méthodes, nous proposons d'ajouter de l'information 3D à l'entrée du réseau de prédiction du visage 3D. Nous ajoutons cette information 3D sous la forme du *Projected Normalized Coordinate Code* (PNCC) [Zhu+16] que nous concaténons avec l'image. Afin de prédire ces PNCC, tout en utilisant le moins de données annotées possible, nous utilisons notre méthodologie et adaptons un réseau génératif à la prédiction de PNCC. Nous avons utilisé l'architecture GMDA-R utilisée précédemment pour la tâche de détection de points d'intérêt faciaux, équipée d'ITL et de connexions directes entre l'encodeur et le décodeur. Grâce à cette architecture, nous avons pu entraîner notre prédicteur de PNCC avec uniquement 50 exemples d'apprentissage. Une fois celui-ci entraîné, nous l'avons utilisé pour annoter un jeu de données de visage. Ensuite, nous avons entraîné une méthode autosupervisée de reconstruction faciale 3D sur ces données en ajoutant le PNCC à l'entrée de son réseau. Nos expériences ont montré que l'ajout du PNCC améliore la pose prédite du visage.

En conclusion, nous proposons une méthodologie, basée sur l'apprentissage par transfert, pour entraîner un réseau de neurones avec peu de données annotées pour certaines tâches de type image-vers-image. Cette méthodologie consiste à adapter un réseau génératif autosupervisé à cette tâche afin de réutiliser les activations de ses couches de

haute dimension. Nous avons appliqué cette méthodologie à deux tâches d'analyse faciale: la détection de points d'intérêt faciaux et la reconstruction de visage 3D. Nos différentes expériences ont montré l'efficacité de notre méthodologie pour ces deux applications.

Table of Contents

Introduction **17**

1 Related work **23**

1.1 Training with limited annotated data . . . 23
1.2 Self-supervised learning for transfer learning . . . 25
1.2.1 Autoencoders . . . 26
1.2.2 Encoder-like models for representation learning . . . 26
1.2.3 Generative models . . . 27
1.2.4 Image-to-image translation tasks . . . 30
1.3 Active learning . . . 38
1.3.1 Presentation . . . 38
1.3.2 Strengths and weaknesses . . . 39
1.4 Face alignment . . . 39
1.4.1 Heatmaps for face alignment . . . 41
1.4.2 Hourglass networks . . . 41
1.4.3 3D face alignment . . . 42
1.4.4 Semi-supervised methods . . . 42
1.5 3D face reconstruction . . . 44
1.5.1 3D Morphable Model . . . 45
1.5.2 Supervised methods . . . 46
1.5.3 Self-supervised methods . . . 48
1.5.4 Hybrid methods . . . 50
1.6 Conclusion . . . 51

2 General methodology **52**

2.1 Introduction . . . 52
2.2 The generative model architecture . . . 53
2.2.1 GMDA-R . . . 53
2.2.2 GMDA-S . . . 54

2.3 Adapting the generative model to the image-to-image translation task . . . 56
2.3.1 Original Interleaved Transfer Layers . . . 56
2.3.2 Two-flow Interleaved Transfer Layers (TF-ITL) . . . 59
2.3.3 Hybrid Interleaved Transfer Layers (H-ITL) . . . 59
2.4 Adding skip-connections (SC) . . . 60
2.5 Conclusion . . . 60

3 Application to face alignment 63
3.1 Application specificities . . . 64
3.1.1 Adapting GMDA to face alignment . . . 64
3.1.2 Active learning for face alignment . . . 64
3.2 Datasets . . . 65
3.2.1 Datasets for 2D face alignment . . . 65
3.2.2 Datasets for 3D face alignment . . . 66
3.3 Experimental settings . . . 66
3.3.1 Model architectures . . . 66
3.3.2 Training . . . 67
3.3.3 Evaluation . . . 68
3.4 Results on 2D face alignment . . . 68
3.4.1 Comparison with state-of-the-art . . . 69
3.4.2 Training with active learning . . . 72
3.4.3 Architecture selection . . . 76
3.5 Results on 3D face alignment . . . 81
3.5.1 Comparison with fully-supervised methods . . . 81
3.6 Conclusion . . . 84

4 Application to 3D face reconstruction 85
4.1 Application specificities . . . 85
4.1.1 Adapting GMDA to 3D face reconstruction . . . 85
4.1.2 The PNCC predictor . . . 86
4.1.3 The 3D face reconstruction model . . . 88
4.2 Experimental settings . . . 88
4.2.1 Training datasets . . . 88
4.2.2 Architectures and training parameters . . . 90
4.2.3 Evaluation metrics . . . 91

4.2.4 Evaluation dataset 91
4.3 Results 92
4.3.1 PNCC prediction 92
4.3.2 3D face reconstruction 92
4.3.3 Head pose rotation estimation 95
4.3.4 Qualitative results 95
4.4 Conclusion 97

Conclusion and perspectives **99**

Bibliography **105**

List of Figures

A Notre méthodologie GMDA pour entraîner un réseau avec peu de données annotées. . . . 6

1.1 An autoencoder for face reconstruction. . . . 26
1.2 Transfer learning from self-supervised learning encoder-like models for supervised tasks. . . . 27
1.3 Contrastive learning principle. . . . 28
1.4 GAN framework. . . . 29
1.5 The StyleGAN generator. . . . 31
1.6 Transfer learning from self-supervised generative models for supervised tasks. 32
1.7 Examples of self-supervised image-to-image translation tasks. . . . 32
1.8 Transfer learning from self-supervised generative models for self-supervised image-to-image translation tasks. . . . 34
1.9 Semantic segmentation: a supervised image-to-image translation task. . . . 35
1.10 Transfer learning from self-supervised generative models for supervised image-to-image translation tasks. . . . 36
1.11 StyleGAN inversion methods. . . . 37
1.12 Facial landmark annotations. . . . 40
1.13 Facial landmark heatmaps. . . . 41
1.14 Skip-connections. . . . 42
1.15 3FabRec pipeline. . . . 44
1.16 The 3D face reconstruction pipeline. . . . 45
1.17 The Normalized Coordinate Code (NCC) and the Projected Normalized Coordinate Code (PNCC). . . . 47
1.18 Illustration of UV position maps. . . . 48
1.19 MoFa architecture. . . . 50

2.1 Our GMDA methodology pipeline to train a network on a task with limited annotated data. . . . 52
2.2 The self-supervised generative model used in our GMDA-R models. . . . 54

2.3 The self-supervised generative model used in our GMDA-S models. 55
2.4 GMDA-R version of the generative model modified for the supervised image-to-image translation task. 57
2.5 GMDA-S version of the generative model modified for the supervised image-to-image translation task. 58
2.6 The original ITL configuration and our proposed two new configurations. . 59
2.7 Our models enhanced with skip-connections between the encoder and the ITLs. 61

3.1 Comparison of heatmap magnitudes. 65
3.2 Face images from WFLW selected with active learning. 75
3.3 Face images from 300-W selected with active learning alongside the ground truth and predicted heatmaps. 76
3.4 Comparison of the failure rates of GMDA-R and GMDA-S. 78
3.5 Comparison of the landmarks predictions of GMDA-R and GMDA-S. . . . 79

4.1 Face images and their PNCC. 86
4.2 Our two-stage framework for 3D face reconstruction training. 87
4.3 Our PNCC predictor architecture. 88
4.4 Our 3D face reconstruction architecture. 89
4.5 Comparison between ground truth and predicted PNCCs. 93
4.6 Comparison of some 3D face reconstruction predictions. 96

List of Tables

3.1 Comparison with face alignment SOTA in the fully-supervised setting on 300-W and WFLW. 70
3.2 Comparison with face alignment SOTA in the fully-supervised setting on AFLW. 70
3.3 Comparison with face alignment semi-supervised methods on 300-W. . . . 71
3.4 Comparison with face alignment semi-supervised methods on WFLW. . . . 72
3.5 Comparison with face alignment semi-supervised methods on AFLW. . . . 73
3.6 Comparison of active learning acquisition functions. 74
3.7 Comparison of our different versions of our architecture. 77
3.8 Impact of encoder fine-tuning. 81
3.9 Comparison with fully-supervised face alignment methods on AFLW2000-3D. 82
3.10 Comparison of sampling methods on AFLW2000-3D. 83

4.1 Dense alignment, 3D face reconstruction and head pose metrics on AFLW2000-3D. 95

INTRODUCTION

Context

The rise of Deep Learning

Artificial neural networks are not a recent invention, the Perceptron [Ros58], usually referred as the first artificial neural network, was proposed in 1958 by psychologist Frank Rosenblatt. However, they were long discarded by the machine learning community who preferred other methods such as Support Vector Machines [CV95]. This changed in 2012 when the neural network AlexNet [KSH12] proposed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey E. Hinton won the 2012 object recognition challenge ImageNet LSVRC by a significant margin. From this moment, a craze for neural networks began in the machine learning community and they quickly spread into many research domains, such computer vision, natural language processing or audio processing, greatly improving the state-of-the-art of these domains.

Artificial neural networks are composed of several layers of neurons put together. The input signal, (i.e. an image for a computer vision task) goes through the layers of the network and is progressively transformed into network features. The last layer outputs the network prediction.

The success of neural networks comes from their ability to learn automatically optimal features. Instead of relying on hand-crafted features such as HOG [DT05] or SIFT [Low04], that might be sub-optimal depending on the task, the first layers of the networks act as the feature extractor whereas the last layers focus on resolving the task using these learned features as input. By stacking more layers, the learned features can model more complex data, improving the ability of the network to resolve complex tasks. Because neural networks gained more and more layers (their depth increasing) over the years, the term *Deep Learning* was coined to refer to artificial neural networks.

The annotation issue

Neural networks are trained by minimizing a training loss (also called cost function) using a training dataset. Traditionally, training is based on the *supervised learning* principle. Supervised learning trains a model to map an input data to a target value. In this case, the training loss is based on the error between the predicted labels and the ground truth labels. Common training losses are the Mean Squared Error for regression tasks and the Cross-Entropy for classification tasks. It is the most straight forward way to train a neural network but it requires *annotated* (we may also use the term *labeled* interchangeably in this manuscript) training data. For example, if you are training a model for image classification, you need training images to classify and for each image, the class of the image. A human is usually required to annotate the training samples.

Bigger neural networks can model more complex functions, thus solving more complex tasks but the larger the network, the more parameters need to be optimized. If the training dataset is not large enough, there is a risk of overfitting. Overfitting happens when a model learns to perform very well on the training data, but its performance on unseen or new data is significantly worse. This can be spotted using a validation set. If the training loss keeps decreasing but the validation loss increases in the meantime, it usually means that the model is overfitting. The model essentially memorizes the training data rather than learning the general patterns that would allow it to make accurate predictions on new data. Large neural networks are more prone to overfitting because they have many parameters that they can adjust to perfectly fit the training data.

While some techniques, such as dropout [Sri+14], regularization or data augmentation, can in some extent prevent overfitting and improve the network performance, the best way is still to increase the training dataset size. However, in the case of supervised learning, gathering large annotated dataset can be difficult. It usually requires an human to annotate the data which can be time-consuming when many sampled need to be annotated. For image classification, the annotation task is rather simple, the human only needs to select the correct class label of the image but it can be more tedious for other tasks. For example, in the case of object detection, the annotations are the bounding boxes and the object class of all the objects of interest in the image. If there are many objects in the image, annotating a single image might require dozens of minutes. Also, annotators are not error-proof, they might miss some objects or assign incorrect class labels which leads to incorrect and noisy annotations. For some tasks, annotating the data necessitate costly material. This is usually the case for tasks where the model must predict 3D data such

as depth estimation or 3D object reconstruction. This requires the use of depth sensor, 3D scanner or multiple cameras. This also imposes a controlled environment which limits the variety of the dataset.

Thus, the lack of large annotated datasets is one of the main limiting factors to the development of Deep Learning in many possible applications. To circumvent this issue, in the recent years, researchers have experimented ways to train neural networks without annotated data. Unlike annotated data, non-annotated data can be easily gathered in large quantities. For computer vision tasks, images can be retrieved from image databases such as Flicker and videos from YouTube. For natural language processing, Wikipedia provides billions of word sentences. Models trained that way are not directly usable but they learn implicit patterns and representations from the training data. They can be then trained again on a specific task using annotated data. Because they have already learned implicit patterns from their previous training, the number of annotated samples needed to train them on the specific task can be greatly reduced compared to a supervised training from scratch. The principle of adapting a model trained on a task to another task is called *transfer learning*.

Our goal and proposed method

While methods training with limited data have emerged in the recent years, most of them perform transfer learning from low-dimensional representations learned during the training on non-annotated data. After this first training, usually, a small network with these learned representations as input is trained in a supervised manner. While this approach is well suited for tasks where the target value is also low-dimensional, such as image classification, it is less effective if the target value is high-dimensional, like an image for example, because the additional network to train is much larger and thus requires more samples to be trained sufficiently.

This CIFRE PhD, in collaboration with the company InterDigital, aims to solve the problem of training neural networks, with limited annotated data, for areas of interest to InterDigital such as facial analysis. Indeed, many tasks in this field suffer from a lack of annotations, which limits the development of deep learning-based methods for these tasks.

In this thesis, we propose another approach, called Generative Model Decoder Adaptation (GMDA) to tackle the issue of training neural networks with constrained anno-

tated data. We propose to use not only low-dimensional representations but also high-dimensional features from unsupervised generative model decoders during transfer learning when the target value is high-dimensional. By doing so, we only need to add a few additional layers to adapt the generative model to the downstream task and thus, we can train with only a few annotated samples. For example, for the face alignment task we trained some of our models with only 50 annotated samples instead of a few thousands and still got decent results. We also demonstrate how the target value of some supervised facial analysis tasks can be transformed into a high-dimensional value which makes our proposed method applicable to these tasks.

Manuscript organization

The rest of this manuscript is organized as follows.

In Chapter 1, we briefly present several learning schemes to train without or with reduced annotated data. Then, we introduce the application of self-supervised learning in the context of transfer learning. We present two distinct model types employed for this purpose: encoder-like models and generative models. Furthermore, we describe the concept of active learning, an alternative approach to minimize the requirement for annotated samples in model training. Subsequently, we narrow our focus to two specific applications: face alignment and 3D face reconstruction. We present existing methods in these domains, with a particular emphasis on techniques that try to achieve effective training using a constrained amount of annotated data.

In Chapter 2, we present our GMDA methodology to train with limited annotated data for supervised image-to-image translation tasks. To do so, we adapt a pre-trained generative model to the image-to-image translation using only a few annotated samples. We present in this chapter, several possible generative models and different ways to adapt the model to the supervised image-to-image translation task with the goal of training with as fewer as possible annotated samples.

Chapter 3 presents the application of our general method to face alignment. We describe how our GMDA methodology can be applied to this task and present the results our experiments for the different generative models and model adaptations, described in Chapter 2, on several face alignment datasets with variable training set size. Our models can be trained with even only 50 samples. We also compare our models to existing face alignment methods, especially the ones who train with limited annotated data. On sev-

eral datasets, our models outperform all of them in multiple low training data settings. Additionally, we propose an active learning scheme to select the best samples to annotate. On some datasets, it makes it possible to halve the amount of training data while still obtaining the same test performance.

In Chapter 4, we apply our general method to the 3D face reconstruction. We use our method to adapt a generative model to make it predict an image-like representation which encodes some head pose and face geometry information, again training it with limited annotated data. Using this adapted model, we annotate a face image dataset with this image-like representation. Then, we modify a self-supervised 3D face reconstruction by adding this image-like representation to the input of the network. The modified architecture predicts better head pose compared to the baseline.

Finally, we conclude this manuscript in the last chapter and propose some possible future work directions.

Chapter 1

Related work

In this chapter, we first present several kinds of learning principles to train neural networks with limited annotated data and why we have chosen to focus on self-supervised learning. Then, we present the use of self-supervised learning for transfer learning, we define and compare two kinds of models used for this task: encoder-like models and generative models, explaining their differences and listing existing methods for both of them. We particularly detail how generative models can be used for image-to-image translation tasks. We also present the principle of active learning, another way to reduce the number of annotated samples needed to train a model. We then focus on our two applications: face alignment and 3D face reconstruction by presenting existing methods, especially those who try to use limited annotated data during the training.

1.1 Training with limited annotated data

In the introduction of this manuscript, we described the principle of supervised learning and its main issue: the need for annotated data. But machine learning is not limited to supervised learning, there are other kinds of learning, which are dedicated to the training of models without or with limited annotated data:

1. **Unsupervised learning**. This type of learning involves training a model on unlabeled data to identify patterns, structures, and relationships in the data without explicit supervision. In unsupervised learning, the model is typically given a task of clustering or dimensionality reduction. Examples of unsupervised learning algorithms include k-means clustering [Llo82] and principal component analysis (PCA).
2. **Semi-supervised learning**. Semi-supervised learning falls in between supervised and unsupervised learning. In semi-supervised learning, a model is trained on a combination of labeled and unlabeled data. The main idea behind semi-supervised learning is that a model can learn more efficiently and accurately when it has access

to both labeled and unlabeled data. The labeled data helps the model learn the correct output for certain inputs, while the unlabeled data helps the model generalize to new and unseen data. When the training on labeled and unlabeled data happens simultaneously, we use, in this manuscript, the term *joint* semi-supervised learning. As common approaches in semi-supervised learning we can cite pseudo-labeling [Lee+13; DY19] which uses predicted labels of unlabeled samples as ground truth labels and consistency regularization which imposes consistency between predictions of augmentations of the same unlabeled sample [BAP14; Hon+18].

3. **Weakly-supervised learning**. This is a type of machine learning technique where a model is trained using only partial or incomplete labels. Unlike supervised learning, where the model is trained using fully labeled data, in weakly supervised learning, the labels provided are noisy or vaguely related to the task. For example, an object detector can be trained using only weak labels such as image-level tags [Zho+16].

4. **Self-supervised learning**. This kind of learning involves training a model on a task that can be automatically generated from the data itself, without the need for human annotations. Self-supervised learning can be seen as a special case of unsupervised learning, where the model is given a pretext task. This can include tasks like reconstructing the input of the model [KW14], predicting the next word in a sentence [Rad+18], filling in a missing word [Dev+18] or image patch [He+22], or solving jigsaw puzzles [NF16]. By solving these pretext tasks, the model can learn useful representations of the input data that can be transferred to other downstream tasks, such as image classification. It can also be used to train generative models like image or text generators. Self-supervised learning followed by supervised learning can be also seen as a form of semi-supervised learning.

Compared to joint semi-supervised learning, self-supervised learning is agnostic to the downstream task since no annotated data is used during the training. Thus, the self-supervised trained model can be fine-tuned for different downstream tasks or on different datasets which reduces the training time and power consumption compared to joint semi-supervised learning where the whole training on unlabeled and labeled samples must be done for each downstream task or dataset.

Compared to weakly-supervised learning, self-supervised learning does not require any kind of annotated, even weakly, data thus the model can be trained on very large unlabeled

datasets and can learn better representations. Also weakly-supervised learning is prone to overfitting if the labels are too noisy [Zha+17; Shu+19].

Because of these several advantages for self-supervised learning, we have chosen to focus our work on this kind of learning to solve the issue of training with limited annotated data.

1.2 Self-supervised learning for transfer learning

Transfer learning is a machine learning technique where a model that has been trained on one task is re-used as a starting point for a new task. Instead of training a model from scratch, which can be time-consuming and requires a large amount of labeled data, transfer learning leverages the knowledge that a pre-trained model has learned from a previous "source" task to accelerate learning on a new "downstream" task or to reduce the number of annotated training samples needed for the learning of the downstream task. The pre-trained model is used as a starting point for a new model, either by using the pre-trained model as a feature extractor or by fine-tuning the pre-trained model on the downstream task. One common example is using a network trained for image classification on the dataset ImageNet [Den+09] for another image-related task.

The source task learning can be supervised, as illustrated previously using the example of image classification. However, this is not a requirement, actually supervised tasks training datasets are always limited because of the need of annotations which can constrain the generalization ability of trained models and thus their performance for transfer learning. On the other hand, self-supervised learning does not rely on labeled data, which allows for the use of much larger training datasets, sometimes containing over a billion samples. Consequently, the generalization performance of self-supervised models can surpass that of supervised models. This leads to a subdomain to machine learning: self-supervised representation learning. This kind of learning aims to learn, in a self-supervised manner, useful features, e.g., interpretable or that can be used for transfer learning.

The two main families of models used for self-supervised representation learning for images are encoder-like models [DGE15; NF16; Che+20a; Che+20b; Car+20; Li+22] and generative models [Dum+17; DKD17; DS19; He+22].

Defining the autoencoder architecture will help us understand these models.

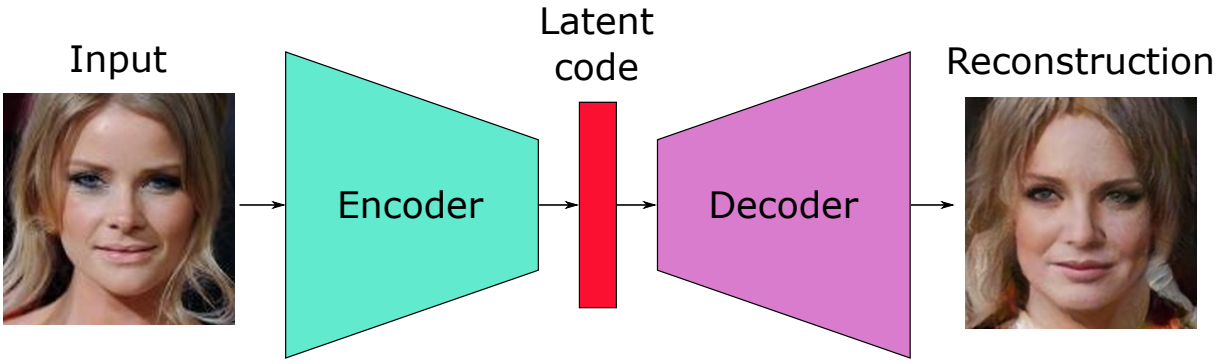


Figure 1.1 – An autoencoder for face reconstruction.

1.2.1 Autoencoders

An autoencoder is composed of two main components: an encoder E and a decoder D. The encoder is a neural network that maps the input data $x \in \mathcal{X}$ to a low-dimensional latent representation $z = E(x) \in \mathcal{Z}$. The decoder is another neural network that maps the latent representation back to the original high-dimensional data $\hat{x} = D(z) \in \mathcal{X}$. The architecture is summarized in Figure 1.1. A traditional autoencoder is trained by minimizing the reconstruction error between the original data and the reconstructed data. As we can see, autoencoders are trained in a self-supervised manner since the pretext task is the training data reconstruction, and does not need annotations.

1.2.2 Encoder-like models for representation learning

In the self-supervised setting, we define encoder-like models as neural networks mainly composed only of an encoder. As in the autoencoder architecture, the encoder takes as input the data and outputs a low-dimensional vector. A small fully connected network is usually added after the encoder to resolve the self-supervised pretext task but is unused for the transfer learning task. Figure 1.2 sums up the process.

Because of the lack of decoder, the network can't be trained using the reconstruction error. Pretext tasks for this kind of models are varied. Doersch et al. [DGE15] predict relative orientation of image patches while Noroozi et al. [NF16] solve jigsaw puzzles of image patches. Many recent methods are based on contrastive instance learning [Che+20a; Che+20b; Car+20; Li+22] which encourages the neural network to learn representations that pull similar instances closer together and push dissimilar instances further apart (see

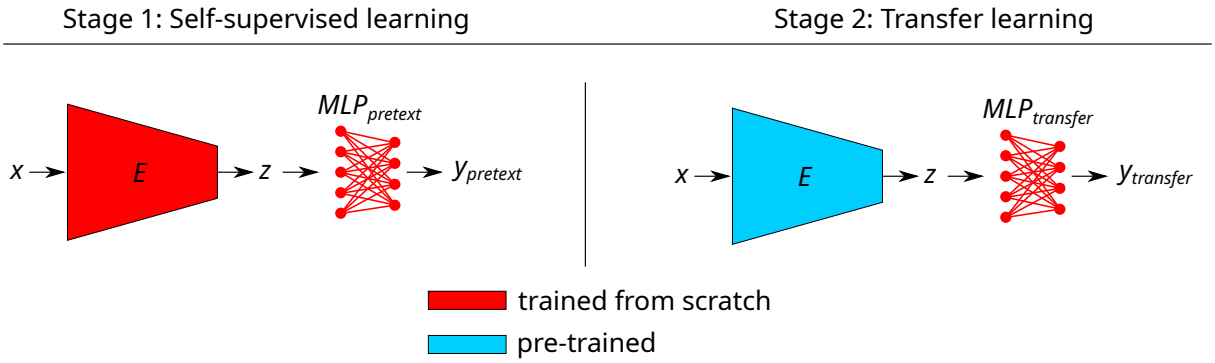


Figure 1.2 – Transfer learning from self-supervised learning encoder-like models for supervised tasks such as image classification or object detection. The pre-trained encoder can be re-used during the transfer learning.

Figure 1.3). These methods can obtain performances close to fully-supervised methods while using only a fraction of the labeled training data.

1.2.3 Generative models

In this thesis, we don't consider the statistical formal definition and simply define generative models as neural networks that can generate new instances of data. A generative model must contain at least a decoder (also called generator), ie, a network which must generate an new instance of data from a low-dimensional input (it can even be just a scalar (seed)). We will see in Section 1.2.3.4 how generative models can be used for transfer learning.

In the domain of face generation, the two main network architectures for face image generation are Variational Autoencoders (VAEs) [KW14] and Generative Adversarial Networks (GANs) [Goo+14].

1.2.3.1 Variational Autoencoders

A Variational Autoencoder (VAE) [KW14] is a type of neural network that can learn to generate new data by encoding and decoding input data such as images.

The VAE is a special kind of autoencoder designed for data generation. A traditional autoencoder minimizes the reconstruction error of the training samples but there is no guarantee that a random vector from the latent space generates a meaningful data when fed to the decoder. Compared to a traditional autoencoder, a VAE introduces a probabilistic component into the model, where the encoder maps the input data to a probability

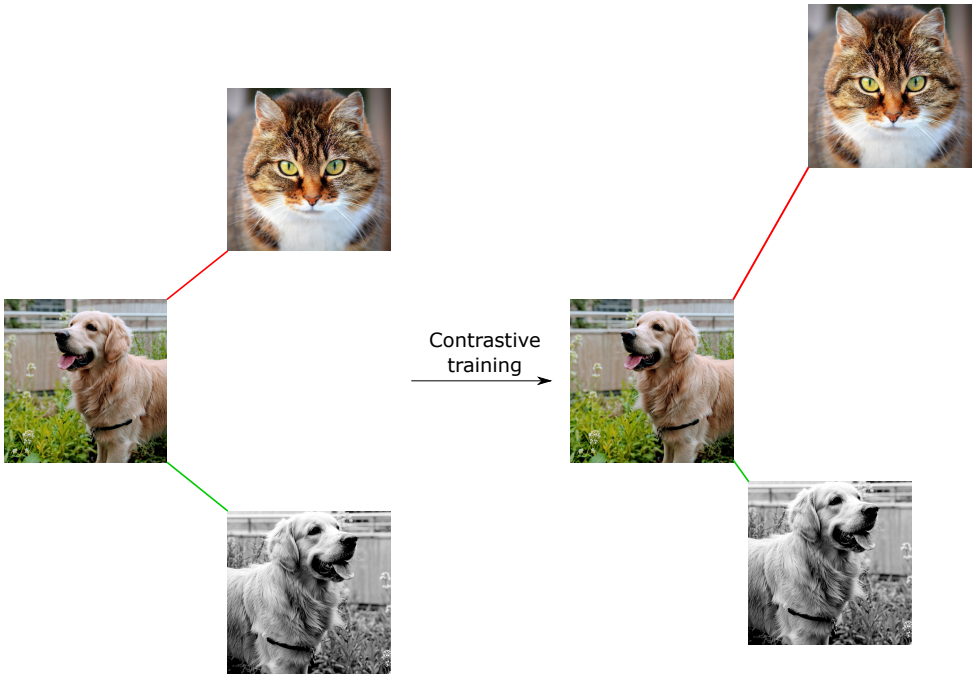


Figure 1.3 – Contrastive training pulls closer representations of transformations of the same image while pushing them apart from representations of different images.

distribution $p(z|x)$ over the latent space rather than to a single vector z, and the decoder samples from this distribution to generate new data. During the training, it minimizes the reconstruction error but also constrains the latent space to have a specific distribution, typically a Gaussian distribution, which encourages the latent space to have a smooth structure that can be easily sampled. However, vanilla VAEs for images, tend to generate images a bit blurry.

1.2.3.2 Generative Adversarial Networks

A Generative Adversarial Network (GAN) [Goo+14] consists of two main components: a generator network G and a discriminator network D. The generator takes a random noise vector z as input and produces a synthetic data point $G(z)$. The discriminator takes either a real data point x or a synthetic data point $G(z)$ as input and produces a binary output indicating whether the input is real or synthetic. The whole architecture can be visualized in Figure 1.4.

During training, the generator tries to produce synthetic data points that are similar to the real data points, while the discriminator tries to distinguish between real and

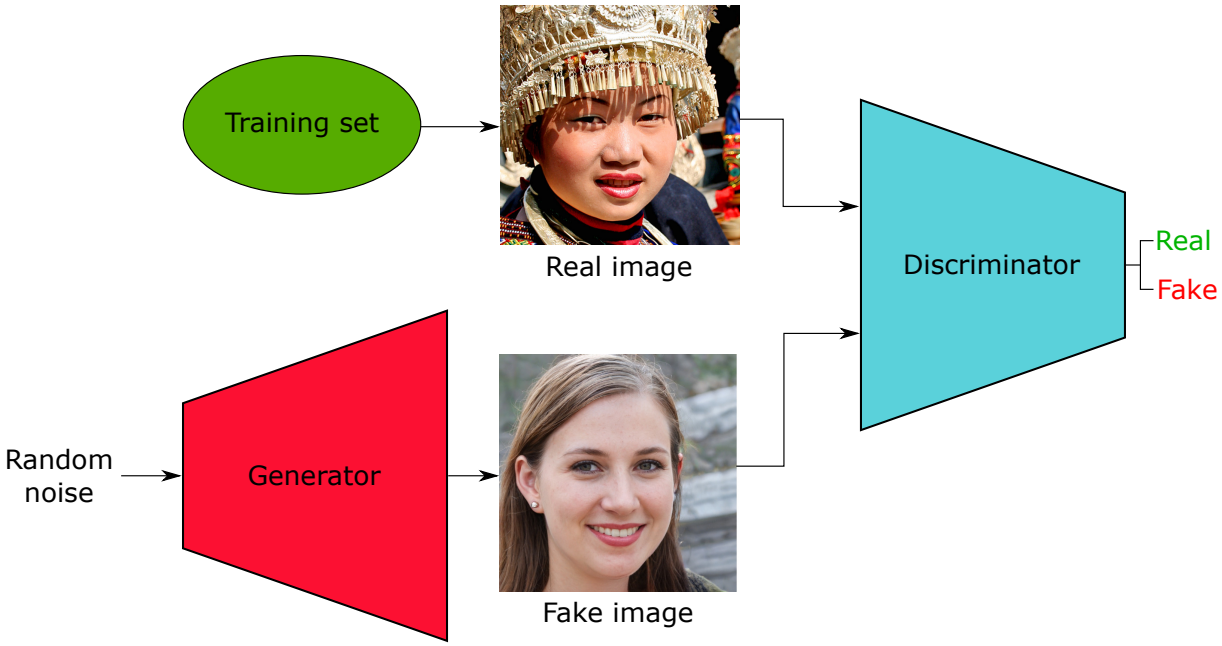


Figure 1.4 – The GAN framework: a generator tries to fool a discriminator with generated images. Both networks are trained concurrently.

synthetic data points. The training is done in an adversarial manner, with the generator and discriminator playing a minimax game. Specifically, the generator tries to maximize the probability that the discriminator will classify its synthetic data points as real, while the discriminator tries to maximize the probability that it will correctly classify real data points as real and synthetic data points as synthetic.

This process continues for many iterations until the generator learns to produce synthetic data points that are indistinguishable from the real data points according to the discriminator. Once the GAN is trained, we can use the generator to generate new data points that are similar to the real data points. To generate a new data point, we simply feed a random noise vector z into the generator and obtain the synthetic data point $G(z)$. In the image domain, the generated images are usually sharper compared to VAEs.

1.2.3.3 StyleGAN

StyleGAN [KLA19; Kar+20; Kar+21] differs from previous GAN architectures by its generative process (see Figure 1.5). Instead of starting from a latent code $z \in \mathcal{Z}$ and progressively increasing the spatial dimensions through the generator layers, z is first projected to an intermediate latent space $\mathcal{W}$ via a mapping network $f : \mathcal{Z} \to \mathcal{W}$ which

produces an intermediate code $w \in \mathcal{W}$. Unlike, $\mathcal{Z}$, $\mathcal{W}$ does not have to support sampling from a fixed distribution (usually Gaussian), its sampling distribution is induced from the mapping network f. This mapping can disentangle the factors of variation from $\mathcal{Z}$ to make them more linear.

Instead of z, the input of the generator is a constant learned vector c_1. At each layer, w is transformed by an affine transformation into a style vector y_i and injected into the current feature map via an Adaptive Instance Normalization (AdaIN) [HB17]. Each style vector controls a specific global (image-level) aspect of the generated image such as the object pose, shape or background. Also, to account for local stochastic variations, random noise is added, pixel-wise, at each layer. Thus the latent space can focus on global factors of variation.

With all these properties StyleGAN can generate varied high quality images.

1.2.3.4 Generative models for representation learning

Initially, GANs can generate images from a latent code but lack a way to obtain the latent code of a real image which make them, at first, inadequate for representation learning. To solve this issue, some methods add an encoder to the GAN architecture to get the latent code of real images [Dum+17; DKD17; DS19] so they can use the model for transfer learning on tasks such as image classification.

Generative models for representation learning are not necessarily convolutional networks. He et al. [He+22] use an autoencoder based on transformers [Vas+17] trained to predict masked image patches. As in convolutional networks, the encoder is then used for transfer learning.

Usually, only the encoder is used for transfer learning so the process is very similar to encoder-like models (see Figure 1.6).

1.2.4 Image-to-image translation tasks

Image-to-image translation is a computer vision domain that involves converting an input image into an output image with a different appearance or style while preserving certain semantic properties. In contrast to image classification or regression tasks, which only output a low-dimensional vector representing the label or target value, image-to-image translation aims to generate a high-dimensional output image that is visually consistent with the input image.

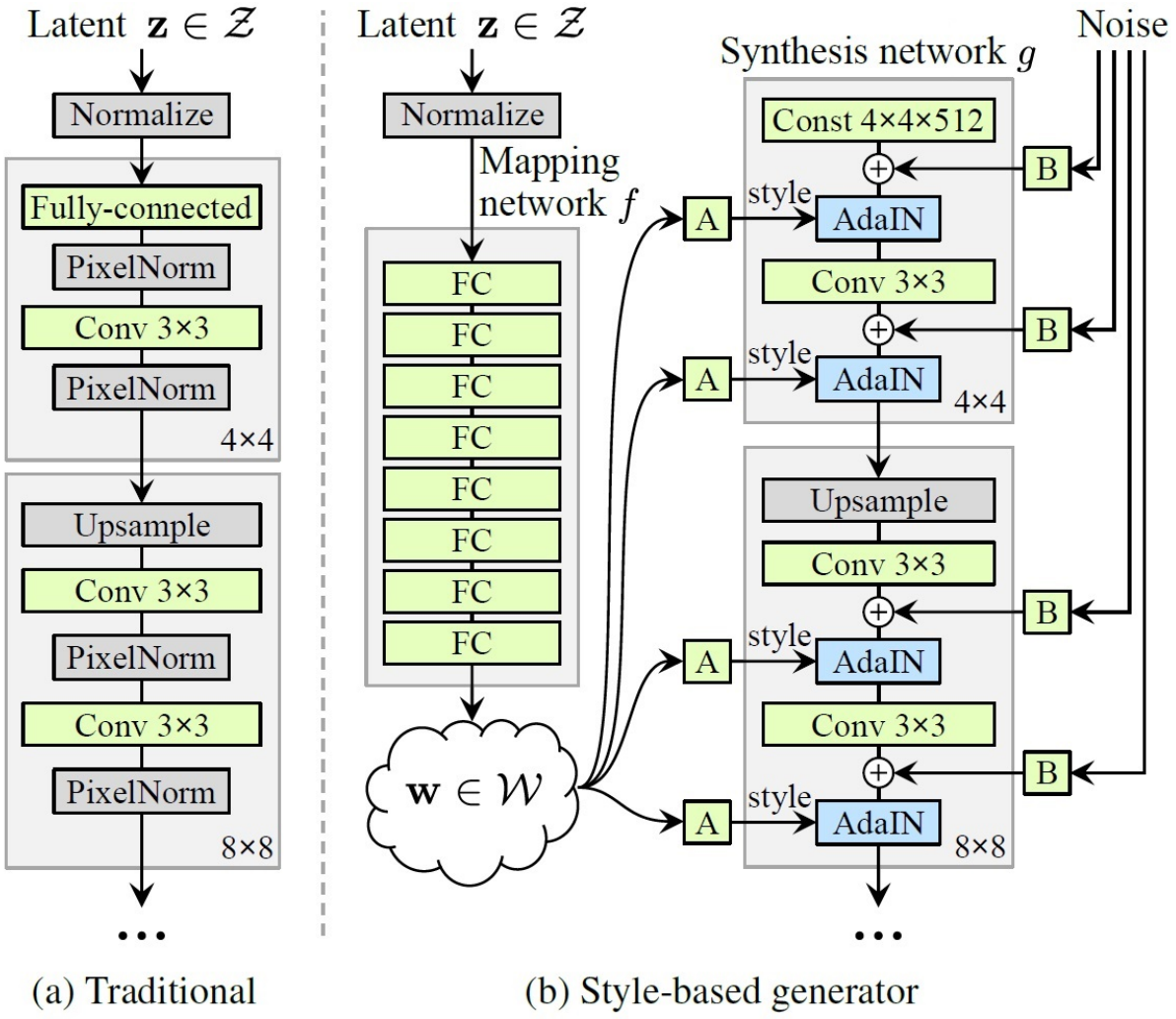


Figure 1.5 – StyleGAN generator (right) compared to a traditional GAN generator (left). Figure from StyleGAN paper © 2021 IEEE [KLA19].

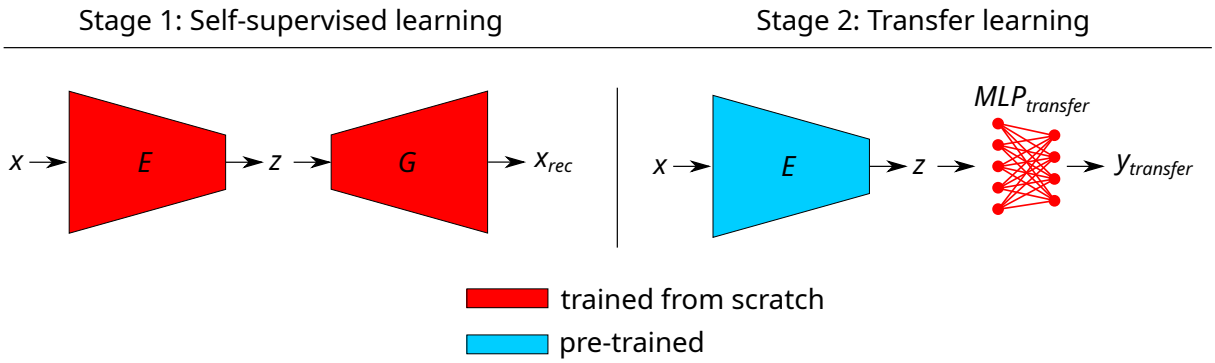


Figure 1.6 – Transfer learning from self-supervised generative models for supervised tasks such as image classification or object detection. Only generative models with an encoder (like VAEs) can be used. The encoder can be re-used during transfer learning but the generator is usually left aside.

Figure 1.7 – Examples of self-supervised image-to-image translation tasks.

Image-to-image translation tasks can be divided into tasks that can be solved with self-supervised learning and tasks that need annotated data.

1.2.4.1 Self-supervised image-to-image translation tasks

For self-supervised image-to-image translation tasks, we can cite inpainting. The goal of inpainting is to remove occlusions from an input image (see Figure 1.7). It can be solved in a self-supervised manner by masking parts of the training images and training the network to fill the gaps using the unmasked images as supervision. Another example of self-supervised image-to-image translation task is super-resolution. The task aims to increase the resolution of input images. Again, it can be trained in a self-supervised fashion by providing downsampled versions of the training images to the network and using the original high-resolution images for supervision.

1.2.4.2 Transfer learning for self-supervised image-to-image translation tasks

While many architectures are specifically designed to solve self-supervised image-to-image translation tasks [Don+15; Led+17; Zhu+17; Sha+20], generative models such GANs or VAEs, initially trained for image reconstruction or generation, can be used for downstream image-to-image translation task such as attribute edition, inpainting or super-resolution [Tov+21; Ric+21; KKC21; Yao+22; Chi+22]. Transfer learning for self-supervised image-to-image translation might seem odd since the downstream task (the image-to-image translation task) is also self-supervised so there is no lack of annotations. However, even self-supervised training can be difficult (time-consuming, unstable, ...) so training from pre-trained weights can speed up the process and even achieve better performance compared to training from scratch.

During the transfer learning, the target images of the self-supervised image-to-image translation task usually lies inside the decoder output distribution but the input data distribution changes. For example, in the case of super-resolution, if the decoder has been trained to generate high-resolution images, the goal is now to find the high-resolution image latent code given only the low-resolution image. The same goes for inpainting, the goal is now to find the latent code of the unmasked image given the masked image. Thus, the decoder does not need to be trained anymore, the task is basically a latent code approximation task. For GANs, an encoder must be trained from scratch to retrieve to latent code [Tov+21; Ric+21; KKC21; Yao+22] but also for VAEs [Chi+22], because the encoder is not adapted anymore to the new task since the input distribution has changed. For example, in the case of super-resolution, the VAE encoder has been trained on high-resolution images but it must now find the latent code of low-resolution images. It might be possible to find a way to re-use the original encoder but the downstream task is self-supervised anyway, so there is no lack of annotated data and it is just simpler to train a new encoder. Figure 1.8 displays the self-supervised and transfer learning (also self-supervised) stages, and which network modules are trained or re-used during each stage.

This kind of transfer learning differs from representation learning previously presented in Section 1.2.2 and Section 1.2.3.4. In these sections, the goal of the self-supervised learning is to train a network to obtain low-dimensional features of the input image which can be then used for downstream tasks such as image classification. These features are obtained through the encoder part of the neural network which is usually almost the whole network for encoder-like models. Even for generative models, once the self-

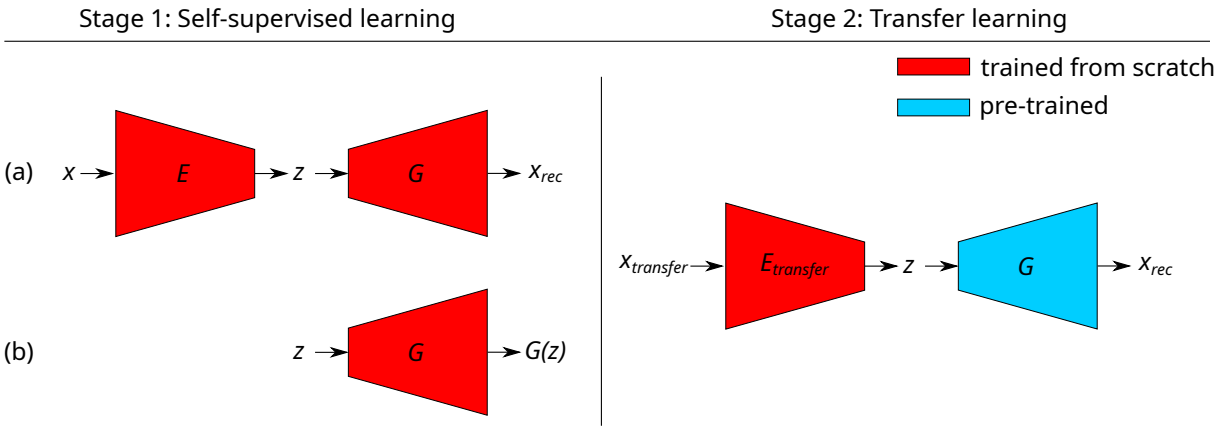


Figure 1.8 – Transfer learning from self-supervised generative models for also self-supervised image-to-image translation tasks such as inpainting or super-resolution. The input data distribution changes so the encoder (if it exists) is not fitted anymore. Hence, a new encoder must be trained but the target image remains inside the generator output distribution so the generator can be re-used without any changes. (a) Generative models with an encoder (i.e. VAEs). (b) Generative models without an encoder (i.e. GANs).

supervised training is done, only the encoder is used for the transfer learning [DKD17; Dum+17; DS19] and the decoder is left aside. It makes sense to only use the encoder for downstream tasks such as image classification since both the encoder output (the latent representation) and the target value (the image label for image classification) are low-dimensional. If the latent representation is of high quality, it is possible to train a small fully connected network to predict the target value.

In the case of transfer learning for self-supervised image-to-image translation tasks, it is usually the opposite, the encoder (if it exists) is left aside and the decoder is re-used.

1.2.4.3 Supervised image-to-image translation tasks

Supervised image-to-image translation tasks are supervised tasks where the target value is an image. An example is semantic segmentation (see Figure 1.9). The goal of semantic segmentation is to assign a object class to each pixel of an input image. The target value is an array of segmentation maps, one for each object class. Thus, the segmentation task becomes is an image-to-image translation task since the target value is an image (without the same number of channels as the input image but with the same spatial dimensions). Another example of supervised image-to-image translation task is

Figure 1.9 – Semantic segmentation: a supervised image-to-image translation task.

face alignment through the use of landmark heatmaps (this will be explained in detail in Section 1.4).

1.2.4.4 Transfer learning for supervised image-to-image translation tasks

Transfer learning from generative models for supervised image-to-image translation tasks is very different from transfer learning for self-supervised image-to-image translation tasks because this time, the input data distribution remains the same but the target image distribution does not lies inside the decoder generative distribution anymore. For example, in the case of semantic segmentation of faces, if we take the example of a VAE which has been trained on a face image dataset, the encoder still receives face images as input but the decoder must now generate segmentation maps instead of face images. Thus, is the decoder of any use for the supervised task or should a new decoder be trained from scratch to generate the target image?

Because the target value is high dimensional, this new decoder can't just be a small MLP like in Section 1.2.2 and Section 1.2.3.4 so its number of parameters to optimize will be high. If the number of annotated training data is constrained, this might lead to poor model performance.

While the target image and the original decoder output are not the same anymore, they still share common information. For example, for semantic segmentation, the face shapes of spatially aligned (face parts are at the same pixel locations for both images), thus it might be possible to re-use some decoder activations to generate the segmentation maps, and thus maybe reduce the number of new parameters to optimize for the new task, but this is not trivial. To the best of our knowledge, the only existing method which uses transfer learning from generative models for a supervised image-to-image translation

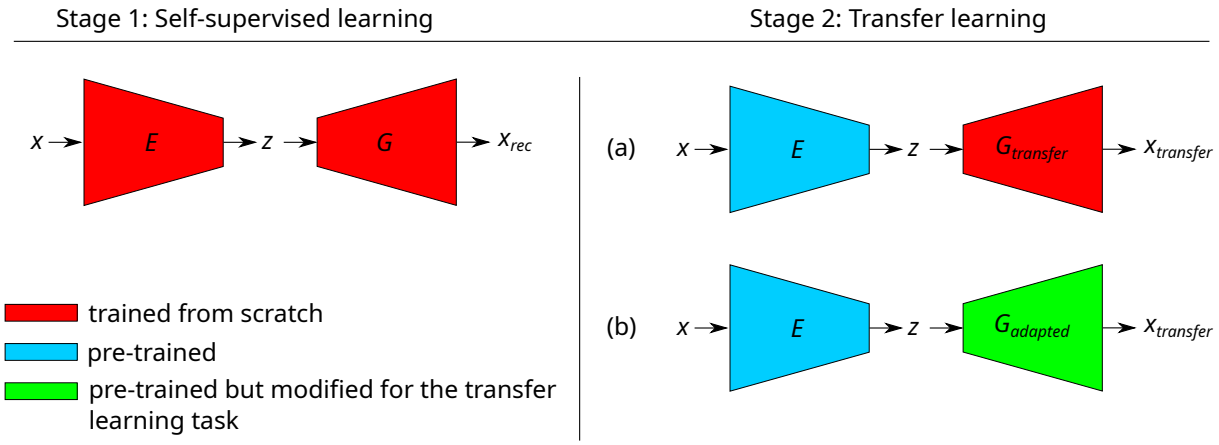


Figure 1.10 – Transfer learning from self-supervised generative models for supervised image-to-image translation tasks. Unlike self-supervised image-to-image translation tasks, this time the target image does not belong inside the generator output distribution so a new generator must be trained (option (a)) or the generator must be modified (option (b)).

tasks, and also re-uses the pre-trained decoder is 3FabRec [BW20] for the face alignment task (detailed in Section 1.4.4). Figure 1.10 shows the possible approaches to perform transfer learning from self-supervised generative models for supervised image-to-image translation tasks.

1.2.4.5 Inverting StyleGAN for image-to-image translation

Because of its strong and disentangled generative power, StyleGAN have been used for image generation but also image-to-image translation tasks such as semantic attribute edition or inpainting [Tov+21; Ric+21; KKC21; Yao+22]. However, StyleGAN lacks an encoder which makes such tasks not trivial.

It is possible to semantically edit a synthetic image generated by StyleGAN by modifying some of its style vectors. However, when editing a real image, we need to first approximate its StyleGAN latent vector, which requires performing a *StyleGAN inversion*. StyleGAN inversion methods can be categorized into three main families: *optimization-based*, *encoder-based* and *hybrid* methods. Optimization-based methods iteratively refine a latent code by minimizing the reconstruction error [GSZ20; KKC21], while encoder-based methods train an encoder to predict the latent code [Tov+21; Ric+21; Xu+21; Nit+22; Yao+22; Wan+22]. Finally, hybrid methods train an encoder to predict an initial latent code which is then refined through optimization [Cha+21]. Although optimization-based

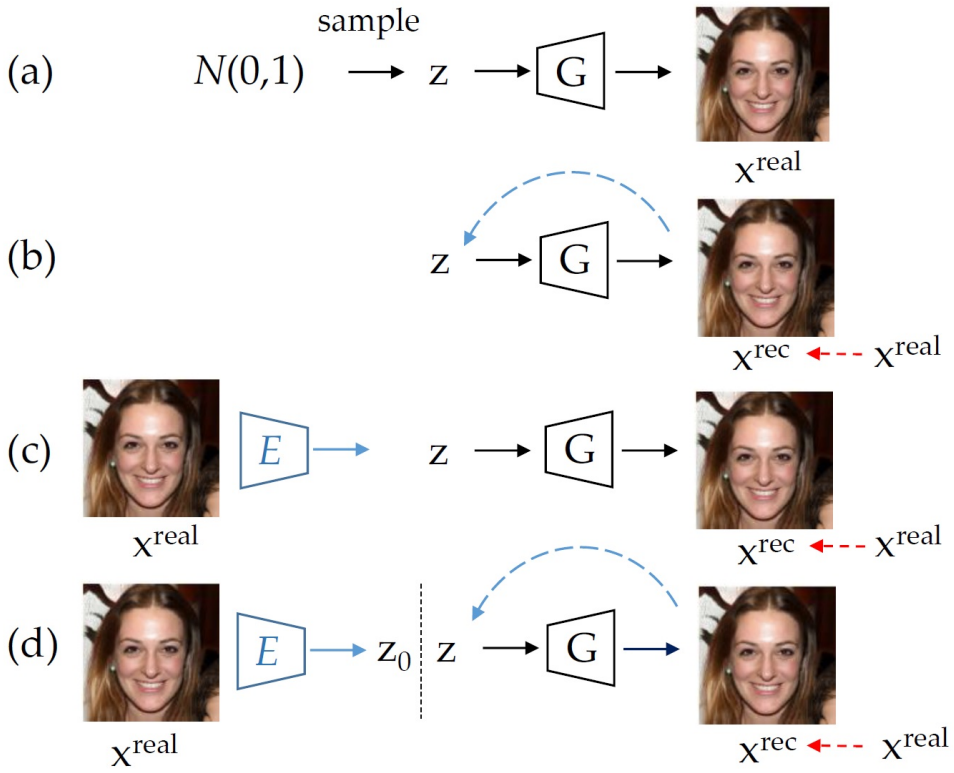


Figure 1.11 – Different kinds of StyleGAN inversion methods. (a) a pre-trained StyleGAN generator can generate faces from random codes. (b) **optimization-based**: inversion is done by iteratively optimizing a latent code to minimize the reconstruction error. (b) **encoder-based**: a network is trained to predict the latent codes of face images. (c) **hybrid-based**: a network is trained to generate an initial latent code which is refined through optimization. Figure from Xia et al. paper © 2023 IEEE [Xia+22].

and hybrid methods generally provide superior results in terms of image quality, they are much slower than encoder-based methods. The different configurations can be seen in Figure 1.11.

Rather than predicting the true latent code z or the intermediate latent code w, many methods predict an extended latent code which contains a different latent code for each style: $w^+ = (w_1, w_2, ..., w_n)$, n being the number of styles [Nit+22; Ric+21]. This extended latent code gives more flexibility and improves the reconstruction quality. Some methods predict along with w^+ a feature map f which replaces the first layers of the generator [KKC21; Yao+22]. This feature map improves the image quality but also makes it possible to encode unaligned images, like rotated or translated images, even though these kinds of images do not exist in the original StyleGAN generative distribution.

1.3 Active learning

1.3.1 Presentation

In academic research, authors typically demonstrate the effectiveness of their proposed method in the settings of low training data by randomly selecting labeled data from the fully annotated training set. But for real-world applications, there may not be labeled data available initially. In such scenarios, carefully selecting which samples to annotate may lead to better trained models, compared to random sampling.

The goal of active learning is to select the best samples to annotate to get the best possible model with a constrained training set. It is very useful when annotating samples is very time-consuming or costly. It follows an iterative procedure described in Algorithm 1. From an unlabeled dataset U and an initial labeled dataset L, the goal is to find the best samples of U to annotate and add to L. A model M is first trained on L, then each sample from U is ranked using an *acquisition function* which depends of the prediction of M for this sample. The top-k samples are annotated, removed from U and added to L. The model is then trained again on the updated L, etc. The procedure goes on until the annotation budget is exhausted.

Algorithm 1: Active Learning process

Input:
The initial labeled dataset L
The unlabeled labeled dataset U
The annotation budget n

Result: The trained model M

$i \leftarrow 0$;
Train a machine learning model M on L;
while $i < n$ **do**
  Use M to predict labels for all unlabeled examples in U;
  Rank each example in U using the acquisition function;
  Select the top-k ranked examples from U;
  Ask an expert to label the selected examples;
  Remove the labeled examples from U;
  Add the labeled examples to L;
  Train M on the updated L;
  $i \leftarrow i + k$;
end

The crucial part of active learning is the choice of the acquisition function. These functions can be categorized into two families, although some methods combine both [KVG19]. The first family is called *uncertainty sampling* [RR08; GIG17; YK19], in this case the acquisition function tries to find the samples where the model is the least confident. It mimics the training loss which is not available for the unlabeled samples. The second family is called *diversity sampling* [SS17], and tries to select samples that represent the diversity of the unlabeled dataset. This approach is particularly useful for classification tasks, where having a class-balanced dataset is important.

1.3.2 Strengths and weaknesses

1.3.2.1 Strengths

As stated previously, active learning can reduce the amount of annotated data needed to train a model by selecting the best samples to annotate, thus reducing the annotation cost. For example, Gal et al. [GIG17] reduce by more than 50% the number of annotated samples need to obtain 5% error rate on MNIST [LeC98]. It can also lead to faster model convergence and improved model performance because the model is trained on the most informative samples. Finally, active learning facilitates human involvement in the machine learning process, allowing domain experts to guide the learning process, correct model errors, and improve model interpretations.

1.3.2.2 Weaknesses

On the other hand, active learning depends heavily on the choice of the acquisition function. An inappropriate one can lead to model bias and poor generalization, for example if the selected samples are too similar. It is an issue that can arise particularly for the acquisition functions based on the uncertainty principle. Also, active learning increases the complexity and the computational overhead of the learning process as the model has to be retrained multiple times.

1.4 Face alignment

The goal of face alignment is to localize specific points on the face, like the mouth corners, the boundaries of the eyes or the tip of the nose. Localizing accurately these keypoints is essential for various applications such emotion recognition or face swapping.

Figure 1.12 – Facial landmark annotations. Top: 2D landmarks. Bottom: 3D landmarks.

However, annotating images for face alignment is time-consuming, especially for images with occlusions or low resolution. Also, some landmarks such as the ones on the outline of the face are ambiguous which may lead to inconsistent annotations among annotators. Supervised face alignment methods need large amounts of training data to achieve good performance in terms of accuracy and generalization. However face alignment datasets, for all the reasons stated above, rarely exceed a few thousand samples making these methods prone to overfitting on the specific training dataset.

Face alignment task involves detecting either 2D or 3D landmarks. 3D landmarks contain depth information but also their 2D position remains at the same anatomical position. For example, for landmarks on the outline of the face, 3D landmarks may be occluded for profile faces whereas 2D landmarks will "slide" to match the visible (but not anatomical) outline of the face. Figure 1.12 shows the differences between 2D and 3D landmarks.

Prior to the development of deep learning techniques in computer vision, face alignment algorithms primarily utilized parametric models such as active shape models [Coo+95] or active appearance models [MB04], or employed cascade regression [Cao+14; Yan+13; XD13]. However, today the vast majority of these methods are based on artificial neural networks.

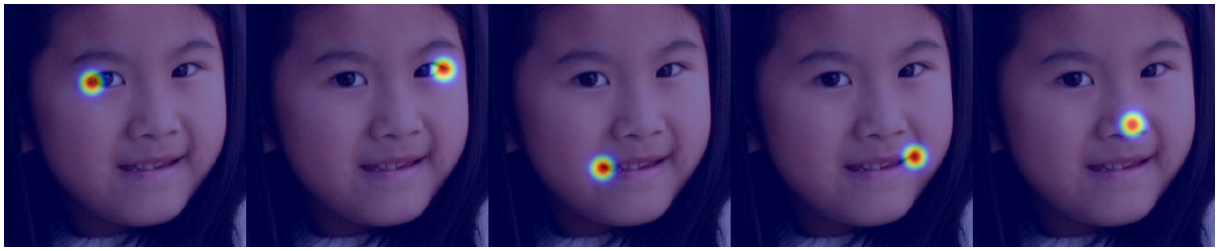


Figure 1.13 – Five heatmaps (overlayed with the face image) corresponding to five different landmarks. The landmark position can be inferred from the heatmap maximum.

1.4.1 Heatmaps for face alignment

Although some recent methods still attempt to directly regress the landmark coordinates [Fen+18b], most methods now utilize heatmap regression [NYD16; BT17b; Wu+18; WBF19; DBC19; Kum+20]. The network is trained to predict a probabilistic heatmap for each landmark (see Figure 1.13), the landmark coordinates can be inferred from the best local maximum. Since heatmaps are images, face alignment can be seen as a supervised image-to-image translation task.

1.4.2 Hourglass networks

A commonly used architecture for face alignment is the *Stacked Hourglass Network* [NYD16] which consists of multiple stacked hourglass modules, each of which is composed of an encoder-decoder subnetwork. The encoder and the decoder in each hourglass module are connected via skip-connections that allow for information to be passed between the encoder and decoder layers (see Figure 1.14). This design allows the network to capture features at different scales and resolutions, which is important for accurately estimating the landmark positions. Each module generates landmark heatmaps that are fed to the next hourglass along with the original image. Stacking the hourglass modules progressively improves the quality of the generated heatmaps.

To improve robustness to large poses and occlusions, Wu et al. [Wu+18] replace landmark heatmaps with facial boundary heatmaps, improving both landmark localization precision and failure rate on multiple datasets. To account for occlusions, Kumar et al. model [Kum+20] landmark uncertainty and visibility as a mixture of random variables, while Zhu et al. [Zhu+19] incorporate weights based on occlusion probability into their model.

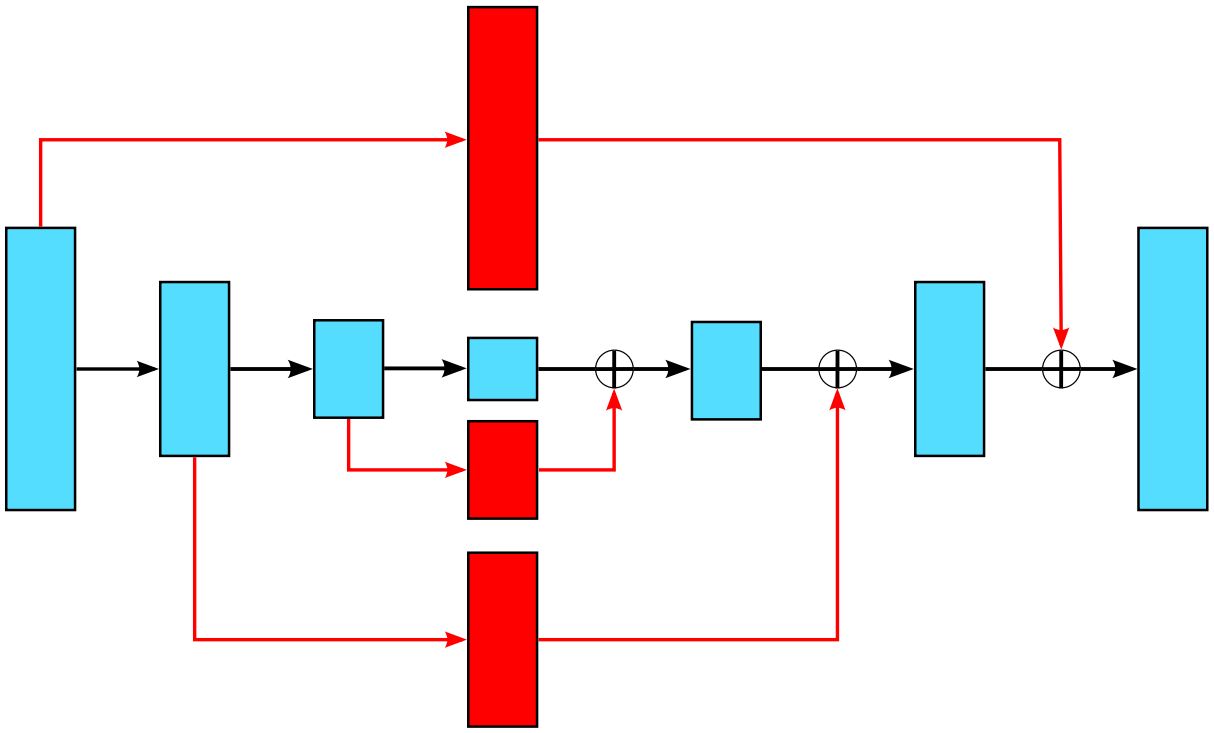


Figure 1.14 – Illustration of skip-connections (in red) in a network.

1.4.3 3D face alignment

For 3D face alignment, some methods try to detect the landmarks directly by predicting heatmaps [BT17b]. But relying only on heatmaps is difficult for occluded 3D landmarks commonly found for profile faces, that's why most methods prefer to fit a 3D face model (this concept will discussed in depth on Section 1.5) [Zhu+16; Guo+20; Rua+21]. Wu et al. [WXN21] learn both tasks simultaneously.

1.4.4 Semi-supervised methods

Semi-supervised methods are used to address the problem of limited annotated training data by incorporating non-annotated into the learning process. To achieve this, various methods have been developed. Qian et al. [Qia+19] generate images with different styles from an input pose image. Honari et al. [Hon+18] enforce the equivariance of landmark predictions over multiple transformations of a face image. Dong et al. [Don+18] transform images into style-aggregated images to deal with the large variance of different image styles. Robinson et al. [Rob+19] generate fake landmark heatmaps from unlabeled images

using a GAN. Dong et al. [DY19] train a teacher to evaluate the quality of student predicted landmarks, and the best samples are added, along with real data, to the next training set for retraining the student detectors.

Some methods are based on transfer learning: VGG-F [Bul+22] trains a neural network, in a self-supervised manner using a contrastive clustering approach [Car+20], on a massive collection of face images. Afterward, the model is adapted by fine-tuning for various facial analysis downstream tasks. FaRL [Zhe+22] leverages masked image modeling and image-text contrastive learning on a large text/image pair dataset to pre-train a network, which can then be utilized for multiple facial downstream tasks.

As stated in Section 1.4.1, predicting landmark heatmaps from a face image can be seen as an image-to-image translation task. 3FabRec [BW20] is also based on transfer learning. It trains an autoencoder to reconstruct face images (self-supervised stage) and then modify its decoder by adding additional convolutional layers, called Interleaved Transfer Layers (ITLs), interleaved with the decoder ones to generate landmark heatmaps instead (supervised stage). Although heatmaps are quite different from face images they still share much information. They can be seen as face images where only information about the face shape as been kept. Thus, their method re-uses both the pre-trained encoder and decoder to generate the heatmaps and can be trained only with a few annotated samples. The supervised stage includes an optional fine-tuning of the encoder and ITLs after training only the ITLs. The 3FabRec framework is presented in Figure 1.15. To the best of our knowledge, it is the only approach based on transfer learning for face alignment which proposes not only a self-supervised learning scheme but also an innovative transfer learning architecture designed to be trained with limited annotated data with the use of the ITLs in the decoder. Other methods based on transfer focus more on the self-supervised learning stage to obtain good representations and use existing convolutional architectures to predict the heatmaps (a simple convolutional network for VGG-F [Bul+22], UpperNet [Xia+18] for FaRL [Zhe+22]).

While 3FabRec architecture can't compete with fully-supervised methods when training with many annotated samples, they are able to train their model with only a few annotated samples. They obtain decent results on many test images even when training with only one annotated sample. On the other hand, for challenging images with low resolution or occlusions, their encoder which is relatively small (Resnet-18 [He+16]) struggles to generate a good latent code which leads to not so accurate heatmaps and thus poor landmarks predictions.

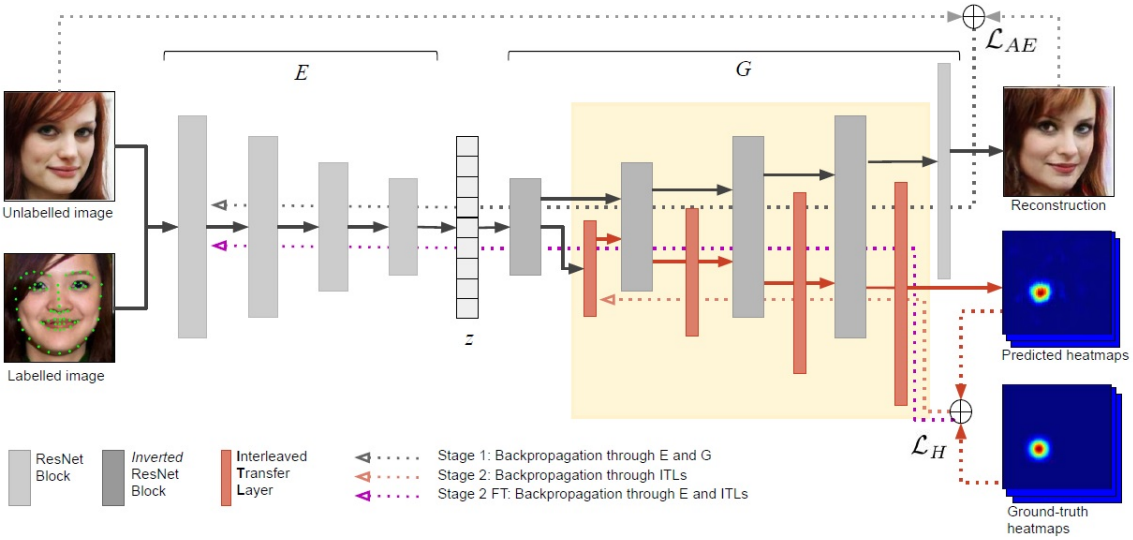


Figure 1.15 – Overview of the 3FabRec pipeline, including the architecture of the autoencoder, as well as the training paths for self-supervised, supervised, and the fine-tuning stages. Figure from 3FabRec paper © 2020 IEEE [BW20].

1.5 3D face reconstruction

3D face reconstruction is the process of creating a three-dimensional digital model of a person's face from two-dimensional images or video footage. The goal is to capture the geometry of a person's face accurately (also the appearance depending on the application). It finds applications in various fields such as facial animation, virtual try-on of cosmetics or accessories, virtual avatars in gaming or other virtual reality applications. In this manuscript, we focus on 3D face reconstruction using only a single monocular face image (see Figure 1.16), so in the rest of this manuscript, every time we refer to the term "3D face reconstruction", this kind of 3D face reconstruction task is implied.

However, getting annotations for 3D face reconstruction is very tedious. It requires a scanner to obtain a 3D face scans so the total number of people in the dataset is very limited. Some datasets are annotated by fitting a face model to an image [Zhu+16]. This makes it possible to obtain datasets with more variety but the annotation quality is quite poor.

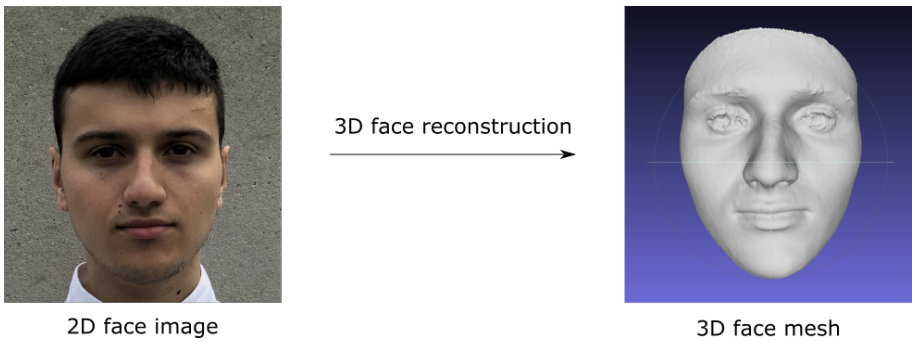


Figure 1.16 – The 3D face reconstruction pipeline.

1.5.1 3D Morphable Model

Most methods for 3D face reconstruction are parametric methods which aim to regress parameters of a 3D Morphable Model (3DMM) [BV03].

A 3DMM is a parametric model that represents a 3D face rig i.e. the possible variations of a 3D face mesh. It can be used to describe the face geometry and color.

1.5.1.1 3DMM for geometry

The 3DMM of face geometry is usually described using two sets of PCA coefficients:

$$\mathbf{S} = \bar{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp} \ , \tag{1.1}$$

where $\mathbf{S} \in \mathbb{R}^{3N}$ is a 3D mesh (a set of N vertices), $\bar{\mathbf{S}} \in \mathbb{R}^{3N}$ the mean face, $\mathbf{A}_{id} \in \mathbb{R}^{3N \times N_{id}}$ is the matrix of the N_{id} principal axes trained on 3D face scans with neutral expression, $\alpha_{id} \in \mathbb{R}^{N_{id}}$ are the corresponding shape coefficients, $\mathbf{A}_{exp} \in \mathbb{R}^{3N \times N_{exp}}$ is the matrix of the N_{exp} principles axes trained on the offsets between expression scans and neutral shapes and $\alpha_{exp} \in \mathbb{R}^{N_{exp}}$ are the expression coefficients.

To align the 3D face with the input view, methods usually predict a 3×3 rotation matrix $\mathbf{T} \in \mathbf{SO}(3)$ (or the pitch, yaw and roll rotation angles), a 2D translation vector $\mathbf{t}_{2d} \in \mathbb{R}^2$ and a scaling factor $f \in \mathbb{R}$. The 3D face can then be projected into the image plane using Weak Perspective Projection:

$$V_{2d}(\mathbf{p}) = f * \mathbf{P_r} * \mathbf{T} * (\bar{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp}) + \mathbf{t}_{2d} \ , \tag{1.2}$$

where $V_{2d}(\mathbf{p}) \in \mathbb{R}^{2N}$ is the 2D projection of the vertices, $\mathbf{P_r}$ the orthographic projection matrix and $\mathbf{p} = (\alpha_{id}, \alpha_{exp}, f, \mathbf{T}, \mathbf{t}_{2d})$.

Compared to directly regressing the face vertices, the advantage of using a 3DMM is that it disentangles face shape and expression so it is easy to generate new plausible faces with different expressions of the same person. The trade off is that face geometry is restricted to the PCA shape and expression spaces so it is hard to regress (or generate) faces with shape or expression very different from the ones of the training 3D scan dataset.

1.5.1.2 3DMM for appearance

A 3DMM can also be used to describe the face skin color. The skin reflectance (also known as albedo) $\mathbf{R} = \{\mathbf{r}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$ is modeled as:

$$\mathbf{R} = \bar{\mathbf{R}} + \mathbf{E}_r \beta \ , \tag{1.3}$$

where $\bar{\mathbf{R}} \in \mathbb{R}^{3N}$ is the average skin reflectance, $\mathbf{E}_r \in \mathbb{R}^{3N \times N_r}$ the matrix of the N_r principal axes of the skin reflectance and $\beta \in \mathbb{R}^{N_r}$ the corresponding PCA coefficients.

To fit the skin color with the one of the image, an illumination model is usually used. The most common is the Spherical Harmonics (SH) model [Mül66].

1.5.2 Supervised methods

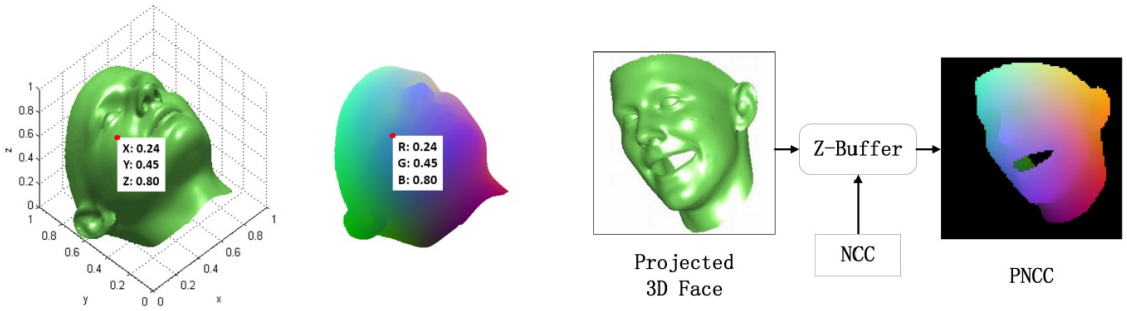
Supervised methods for 3D face reconstructions aim to recover the 3D face geometry (a 3D face mesh composed of vertices) from an input face image. They are trained using annotations provided by the training dataset.

Some methods predict the 3DDM parameters from which they can recover the face mesh. 3DDFA [Zhu+16] uses a cascaded framework, it trains a neural network to predict the parameter update $\Delta\mathbf{p}^k$ from the concatenation of the input face image $\mathbf{I}$ and a Projected Normalized Coordinate Code (PNCC) derived from the current parameter estimate $\mathbf{p}^k$:

$$\Delta\mathbf{p}^k = Net^k(\mathbf{I}, \text{PNCC}(\mathbf{p}^k)) \ . \tag{1.4}$$

The PNCC is based on the Normalized Coordinate Code (NCC). The NCC is the 3D mean face normalized to [0-1] in x, y, z axis:

$$\text{NCC}_d = \frac{\bar{\mathbf{S}}_d - \min(\bar{\mathbf{S}}_d)}{\max(\bar{\mathbf{S}}_d) - \min(\bar{\mathbf{S}}_d)} \quad (d = x, y, z) \ , \tag{1.5}$$



(a) NCC (b) PNCC

Figure 1.17 – The Normalized Coordinate Code (NCC) and the Projected Normalized Coordinate Code (PNCC). (a) The normalized mean face, which is also displayed with NCC as its texture (NCC_x = R, NCC_y = G, NCC_z = B). (b) The generation of PNCC: The projected 3D face is rendered by Z-Buffer with NCC as its colormap. Figure from 3DDFA paper © 2016 IEEE [Zhu+16].

where $\bar{\mathbf{S}}$ is the mean shape of the 3DMM. Since the NCC has three channels like RGB, it can be seen as texture. The PNCC is rendered by coloring the projected vertices of a model with parameter $\mathbf{p}$ with the NCC colormap. The projection is done using the Z-buffer algorithm [Str74] (see Figure 1.17).

$$\begin{aligned} \text{PNCC}(\mathbf{p}) &= \textit{Z-buffer}(V_{3d}(\mathbf{p}), \text{NCC}) \ , \\ V_{3d}(\mathbf{p}) &= f * \mathbf{T} * \mathbf{S} + [\mathbf{t}_{2d}, 0]^T \ , \\ \mathbf{S} &= (\bar{\mathbf{S}} + \mathbf{A}_{id}\alpha_{id} + \mathbf{A}_{exp}\alpha_{exp}) \ . \end{aligned} \tag{1.6}$$

As training losses, the network uses the error between the groundtruth and predicted vertices positions (Vertex Distance Cost, VDC) and the error between the ground truth and predicted 3DMM parameters (Paramter Distance Cost, PDC).

The training of the network is later improved in 3DDFA-V2 [Guo+20] by adding a meta-joint optimization of the VDC and PDC, and also by using landmark regularization.

SADRNet [Rua+21] regresses both a pose-dependent 3D face and a pose-independent 3D face which are aligned to get the final 3D face. It also handles occlusions using attention maps. SynergyNet [WXN21] trains a network to predicts 3DMM parameters from a face image but also another network which predicts 3DMM parameters from 3D face landmarks to add a consistency loss in the training process.

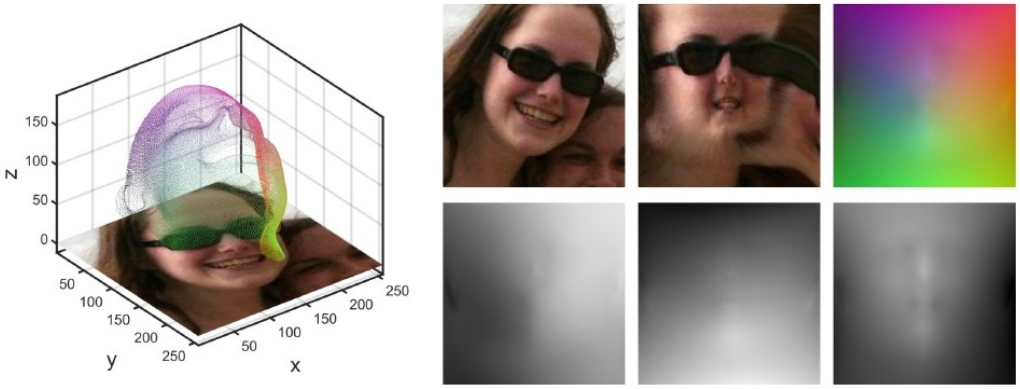


Figure 1.18 – The illustration of UV position maps. Left: 3D plot of input image and its ground truth 3D point cloud. Right: The first row is the input 2D image, extracted UV texture map and corresponding UV position map. The second row is the x, y, z channel of the UV position map. Figure from PRN [Fen+18a] paper, reproduced with permission from Springer Nature.

Not all methods are based on 3DMM. PRN [Fen+18a] encodes a face geometry into a UV position map. Each vertex of the 3D face is assigned a pixel in the UV map and the color of the pixel is the normalized 3D position of the vertex (see Figure 1.18). It trains a network to predict the UV position map from an input image. From the predicted UV map, the face geometry can be inferred. Jung et al.[JOL21] use a Free-Form Deformation model which encodes the 3D face into a set of controls points.

Most of the supervised methods train on datasets with low quality 3D annotations, notably the 300-W-LP dataset [Zhu+16], which extends the 300-W dataset [Sag+13] with larger face poses using pixel warping and uses 3DMM fitting [RV05] to obtain 3D annotations. Thus, the annotations are not so accurate since they are obtained through optimization. Consequently, the predictions of the supervised methods are also not so accurate.

1.5.3 Self-supervised methods

Self-supervised methods don't have access to ground truth 3D face geometry during training, so they mostly rely on image reconstruction loss. They need to model the whole face (geometry and appearance) and project the rendered face in the image space to compare it with the input image. Landmark regularization is also usually used: sparse 2D

landmarks inferred from the predicted 3D face mesh are compared to landmarks predicted by a pre-trained landmark detector.

MoFa [Tew+17] uses a convolutional network to predict the whole 3DMM face and scene parameters from an input face image. The face parameters are the PCA coefficients for the shape and expression: α_{id} and α_{exp} and the PCA coefficients for skin reflectance: β. They render the scene using a pinhole camera model under a full perspective projection $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$. They predict the position and orientation of the camera in the world through a rigid transformation parameterized with rotation $\mathbf{T} \in \mathbf{SO}(3)$ and a global translation $\mathbf{t} \in \mathbb{R}^3$. The functions $\Phi_{\mathbf{T},\mathbf{t}} = \mathbf{T}^{-1}(\mathbf{t} - \mathbf{v})$ and $\Pi \circ \Phi_{\mathbf{T},\mathbf{t}}(\mathbf{v})$ map a point $\mathbf{v}$ from the world to the camera space and then to screen space. As illumination model, they use Spherical Harmonics (SH). They assume the illumination as distant and low-frequency, and skin as a Lambertian surface. Thus, the radiosity at vertex $\mathbf{v}_i$ with a surface normal $\mathbf{n}_i$ and a skin reflectance $\mathbf{r}_i$ is:

$$C(\mathbf{r}_i, \mathbf{n}_i, \gamma) = \mathbf{r}_i \cdot \sum_{b=1}^{B^2} \gamma_b \mathbf{H}_b(n_i) \ . \tag{1.7}$$

The $\mathbf{H}_b : \mathbb{R}^3 \to \mathbb{R}$ are SH basis functions and the $B^2 = 9$ coefficients $\gamma_b \in \mathbb{R}^3$ ($B = 3$ bands) parameterize colored illumination using the red, green and blue channel. Thus, the whole semantic code vector to predict to render the reconstructed face is $\mathbf{p} = (\alpha_{id}, \alpha_{exp}, \beta, \mathbf{T}, \mathbf{t}, \gamma)$.

MoFa uses a differentiable renderer to render the reconstructed face so that they can backpropagate gradients for losses based on the reconstructed image. As losses, they use a photometric loss which compares the rendered image with the input image, a landmark loss which compares 2D landmarks inferred from the predicted mesh to landmarks predicted with a pre-trained landmark detector and a L2 regularization loss on the predicted 3DMM coefficients (shape, expression and skin reflectance) to encourage the network to predict faces that stay close to the average face. MoFa framework can be visualized in Figure 1.19.

RingNet [San+19] builds on MoFa but with some improvements. They use a better 3DMM: FLAME [Li+17], which allows to model more varied facial shapes and expressions. They also impose consistency on the predicted shape 3DMM coefficients α_{id} for images of the same person during training. Deep3DFaceReconstruction [Den+19] adds a perceptual loss which compare features from a pre-trained face recognition network for the input and reconstructed image. They also use a skin detector to compute the photometric loss only

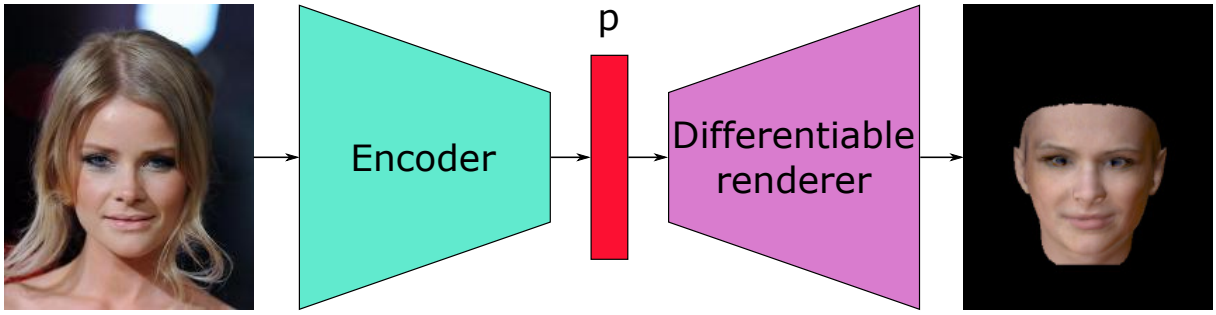


Figure 1.19 – MoFa architecture [Tew+17]. The encoder predicts the semantic code vector **p** of an input face image. From this code, the differentiable renderer renders the reconstructed face.

on skin pixels. DECA [Fen+21] predicts person specific detailed shape in addition to the classical 3DMM parameters to improve the reconstructed shape. FOCUS [Li+23] learns jointly face reconstruction and segmentation to better handle occlusions.

Most of the methods use a ResNet50 [He+16] as encoder. The evaluation used to be qualitative because of the lack of datasets with quality ground truth. Nowadays, the standard evaluation dataset for quantitative evaluation is the NoW dataset [San+19]. This dataset only includes validation and test splits so the lack of quality annotations for training is still an issue. Also it only evaluates the predicted face shape, not the expression.

1.5.4 Hybrid methods

An issue with self-supervised methods is that their training does not take into account the scale the 3D face. A large face far away from the camera would appear the same in a 2D image as a small face close to the camera. Because they don't have access 3D data during training, they can't learn to predict metrical faces (faces with the real scale), they also tend to predict faces with incorrect head pose. MICA [ZBT22] notices this issue and proposes an hybrid method. A pre-trained face recognition network is adapted and trained in a supervised manner on 3D data to predict the 3DMM shape coefficients (only shape, not expression) from a face image. Then, they use Analysis-by-Synthesis from Thies et al. [Thi+16], a self-supervised optimization scheme, to predict the other face and scene parameters. Their framework is designed for 3D face reconstruction on videos but can also be used on still images.

This architecture improves the estimation of face scales compared to self-supervised methods. However the training of the shape predictor requires a lot of 3D annotated data.

1.6 Conclusion

In this chapter, we have seen that to reduce the number of annotated samples needed to train a model, self-supervised learning followed by transfer learning is a possible solution. We have presented two types of models for self-supervised learning: encoder-like models which focus on learning a good low-dimensional representation of the data using pretext tasks such as contrastive learning, and generative models which also include a decoder and are trained mostly by reconstructing the input data. Encoder-like models are efficient for downstream tasks which expect a low-dimensional target value (such as image classification), but in the case of supervised image-to-image translation tasks where target value is high-dimensional, using only the low-dimensional representation as input requires to train a whole new convolutional network which could prove difficult if the available annotated data is constrained.

Methods using transfer learning from generative models for image-to-image translation tasks exist but most of them focus on self-supervised image-to-image translation tasks where they can re-use the decoder because the target image distribution (of the new image-to-image translation task) remains inside the decoder output distribution. In the case of supervised image-to-image translation tasks, the target image distribution changes so the decoder can't be used directly. However, finding a way to re-use this decoder could be an effective way to avoid the need to train from scratch a new decoder during the supervised stage, and thus an effective way to reduce the number of annotated samples needed to train the model. We only found one method doing it, in the application of face alignment. We think that this approach can be extended to other supervised image-to-image translation tasks, that different generative models can used, and that the way they adapt the decoder to the supervised task can be improved.

We also presented the active learning principle which is a effective way to reduce the need of annotated samples, and that could be applied to image-to-image translation tasks.

Finally, we presented two possible applications, face alignment and 3D face reconstruction, where annotations are scarce and may benefit from transfer learning schemes to alleviate the lack of annotations.

Chapter 2

General methodology

2.1 Introduction

In this thesis, we propose a methodology, called **Generative Model Decoder Adaptation (GMDA)**, to train a supervised image-to-image translation task with few annotated data. To do so, we re-utilize not only the latent representation but also the decoder layers of a pre-trained self-supervised generative model to generate the target image. Unlike methods which only use the latent representation, this methodology requires to modify the decoder to adapt it to the supervised image-to-image translation task. This principle is inspired from 3FabRec [BW20] which uses it for face alignment but in this thesis, we generalize this principle to other image-to-image translation tasks. We study two generative models, the one used by 3FabRec but also another one using a StyleGAN [KLA19; Kar+20; Kar+21] encoder and decoder. We also propose other ways to adapt the decoder to the supervised image-to-image translation task. Finally, we also propose to add skip-connections between the encoder and the decoder to improve the network precision.

Our method pipeline can be seen in Figure 2.1.

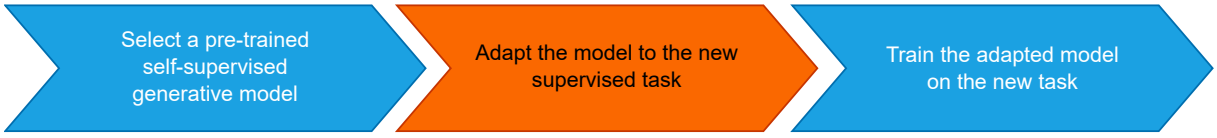


Figure 2.1 – Our GMDA methodology pipeline to train a network on a task with limited annotated data.

2.2 The generative model architecture

The generative model architecture should follow several criteria to maximize the transfer learning performance.

1. The generative model should already have a pre-trained encoder so we don't need to train it from scratch during the supervised training.
2. The output of the encoder (the latent representation) should contain enough information for the downstream task.
3. Since we would like to perform the transfer learning with limited annotated data, the architecture should be easy enough to adapt to the downstream task without the need to add many new layers or fine-tune a lot of pre-trained parameters during the supervised training.

We study two possible architectures for the generative model: an autoencoder based on the ResNet architecture [He+16] and a model based on StyleGAN [KLA19]. Both architectures used convolutional layers but are still quite different making them interesting cases to study. The next paragraphs explain their differences and why we choose these two architectures.

2.2.1 GMDA-R

Our first generative model studied is the one proposed by 3FabRec [BW20]. It is an autoencoder with a ResNet-18 [He+16] as encoder and an inverted ResNet, using deconvolutions, as decoder. Thus, we call this version of our methodology: GMDA-R (R for ResNet). The autoencoder is trained to reconstruct face images in a self-supervised manner using millions of face images. Similarly to 3FabRec, we also use a GAN [Goo+14] discriminator and latent code regularization [Mak+16] to improve the reconstruction and smooth the latent space. This GAN discriminator is only used during the self-supervised training and is discarded afterwards. The training face images are very varied thus the latent representation is robust to many factors such as face pose, age or skin color. The architecture can be seen in Figure 2.2

3FabRec gets good results with this architecture for the face alignment task but the generative model has several limitations. The quality of the face reconstruction or generation is mediocre. This is due to the fact that the generative model architecture (based on ResNet18) is quite small compared to current generative models [KLA19; Kar+20;

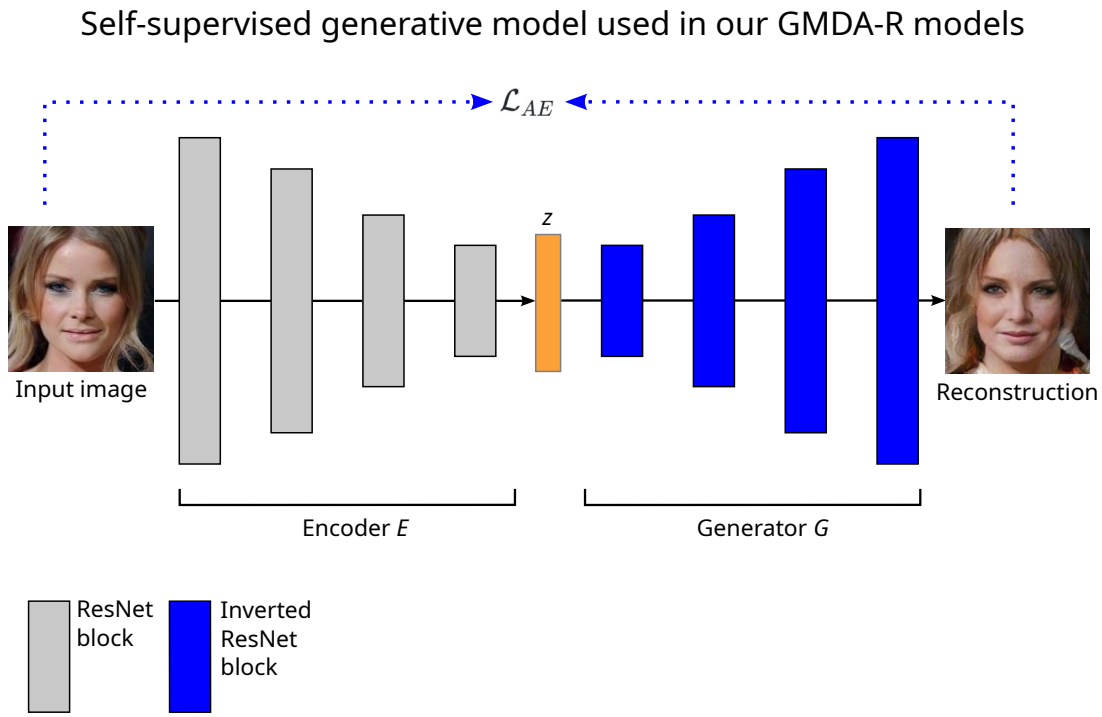


Figure 2.2 – The self-supervised generative model, based on the ResNet autoencoder, used in our GMDA-R models. Similar to the one used by 3FabRec [BW20].

Kar+21; SSG22]. Also, the latent code dimension is only 99. While our primary goal is not the face reconstruction quality but maximizing the performance of the supervised downstream task with as few annotated data as possible, the former may be correlated to the latter.

2.2.2 GMDA-S

We try to alleviate the possible weaknesses of the previous architecture by using the state-of-the-art face generator: StyleGAN [KLA19; Kar+20; Kar+21] as our decoder. This network has a much bigger capacity compared to a ResNet18 and the latent code, of size 512, is introduced not at the start of the generator but at each layer block of the generator (see Section 1.2.3.3). It obtains spectacular results for face generation, capturing both local and global details. It has already been used for image-to-image translation tasks (see Section 1.2.4) so we test how well this architecture works in our framework. Since we don't want to train an encoder during the supervised stage (as discussed at the beginning of this section), we also use a pre-trained StyleGAN encoder.

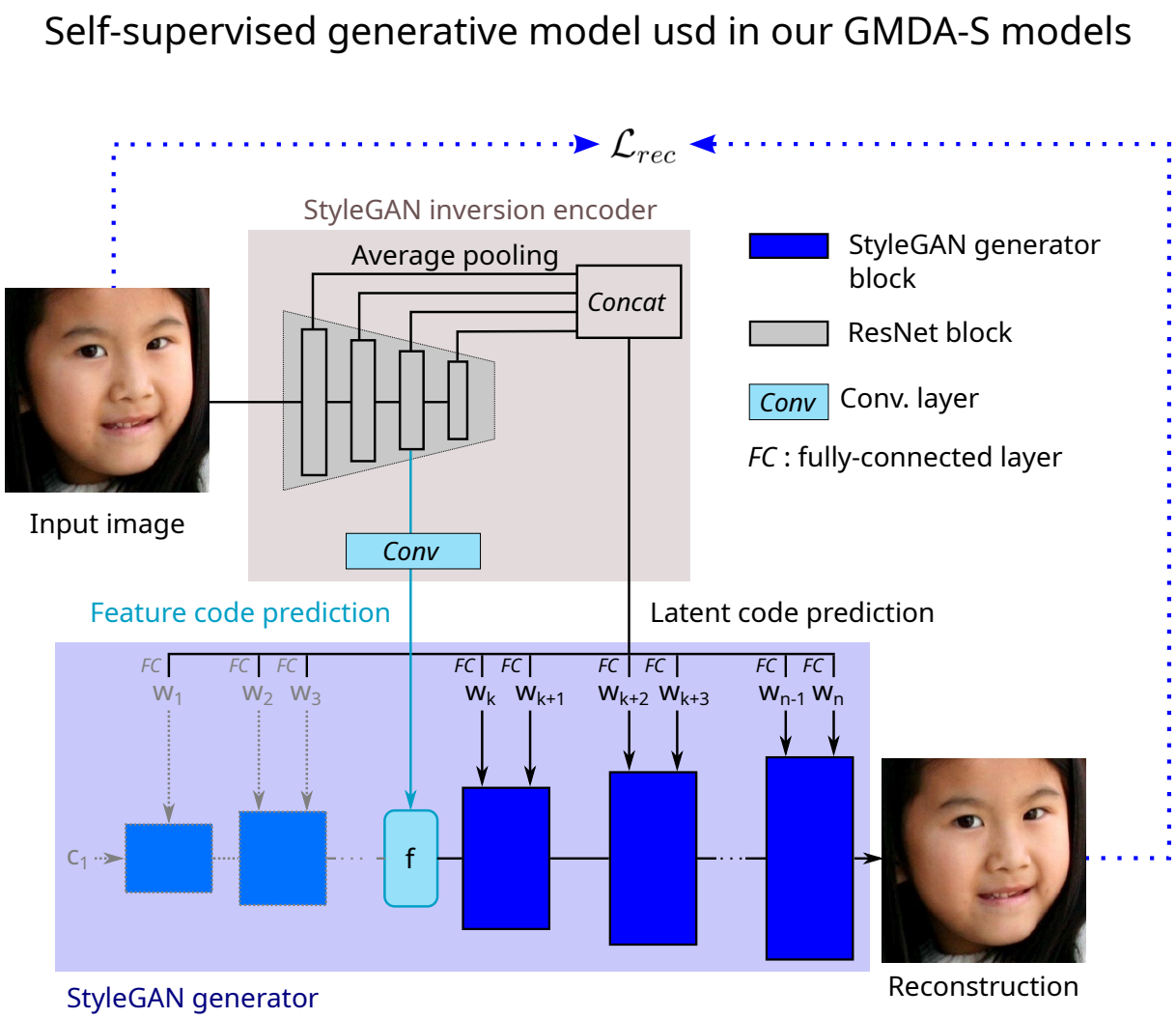


Figure 2.3 – The self-supervised generative model, based on StyleGAN, used in our GMSA-S models. This StyleGAN encoder [Yao+22] predicts 2 codes: a feature code which replaces the first layers of the generator, and the latent code. From these 2 codes, the generator reconstructs the face image.

Because the images used for our different downstream tasks do not necessarily follow the alignment used to train the StyleGAN generator, we need an encoder which can encode unaligned images. That's why we use the Feature-Style encoder [Yao+22]. This encoder predicts 2 codes: the feature code f and the extended latent vector w^+ (of size 18×512). During image generation, the first $k-1$ layers of the generator are replaced by the feature code f. Then, each latent vector w_i ($i \geq k$) from w^+ is transformed into a style vector by an affine transformation and injected into the corresponding StyleGAN layer through AdaIN. The last block outputs the reconstructed image.

This new architecture has a better face reconstruction results and a bigger latent representation, however it is more complex than the ResNet and a bigger capacity means more parameters to optimize if we fine-tune the encoder during the supervised training. The architecture is displayed in Figure 2.3. We call our models based on StyleGAN GMDA-S (S for StyleGAN).

2.3 Adapting the generative model to the image-to-image translation task

We need to adapt the generative model to the downstream image-to-image translation task so that it generates our target image-like value instead of the reconstruction of the face image. The adaptation should add as fewer new parameters to optimize as possible to make it easier to train it with few annotated data. Since the target is an image closely related to the reconstructed image (the output of the decoder), we would like to re-use the generative power of the pre-trained decoder for our new task. This means finding a way to re-use the decoder layers activations.

2.3.1 Original Interleaved Transfer Layers

Our baseline to adapt the generative model uses the Interleaved Transfer Layers (ITLs) from 3FabRec [BW20]. An ITL is a convolution layer located after a decoder block. It takes as input the the output of the decoder block and, apart from the last ITL, outputs a new feature map with the same spatial dimensions and channel number of the input. This new feature map is fed to the next decoder block. The last ITL generates the target image. When generating this target image, the latent representation from the encoder

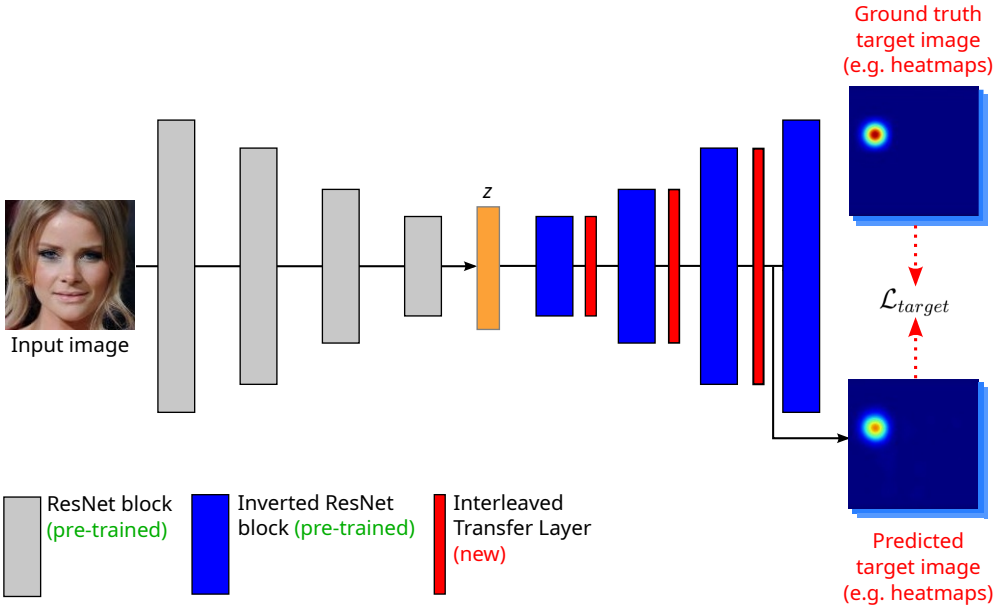


Figure 2.4 – GMDA-R version of the generative model modified for the supervised image-to-image translation task. The encoder and generator layers are already pre-trained from the self-supervised training of the generative model. Interleaved Transfer Layers (ITLs) are added between the decoder layers and are trained in a supervised manner to generate the target image of the supervised image-to-image translation task. Here, target images are landmark heatmaps but the method can be applied to other tasks. In the case of landmark heatmaps, this architecture is the same as 3FabRec [BW20].

goes through both the decoder layers and ITLs. The ITLs re-use the activations of the decoder layers but adapt them to the target image-to-image translation task.

The GMDA-R and GMDA-S architectures with the added ITLs can be seen in Figure 2.4 and Figure 2.5 respectively.

The ITLs satisfy the conditions enumerated previously in Section 2.2 and 3FabRec has proved their efficiency for training with limited data on the face alignment task. However their configuration might still restrict the ability of the network to adapt to the downstream task. Indeed, each ITL, apart from the last one, has two tasks to perform, retrieve the useful information for the downstream task from the previous decoder layer output while still maintaining a meaningful input for the next decoder layer. One could try to fine-tune the decoder layers alongside training the ITLs to remove pressure on the ITLs but it increases the number of parameters to optimize and our experiments showed that it actually hurt the model performance.

Adaptation and supervised training of the generative model used in our GMDA-S models

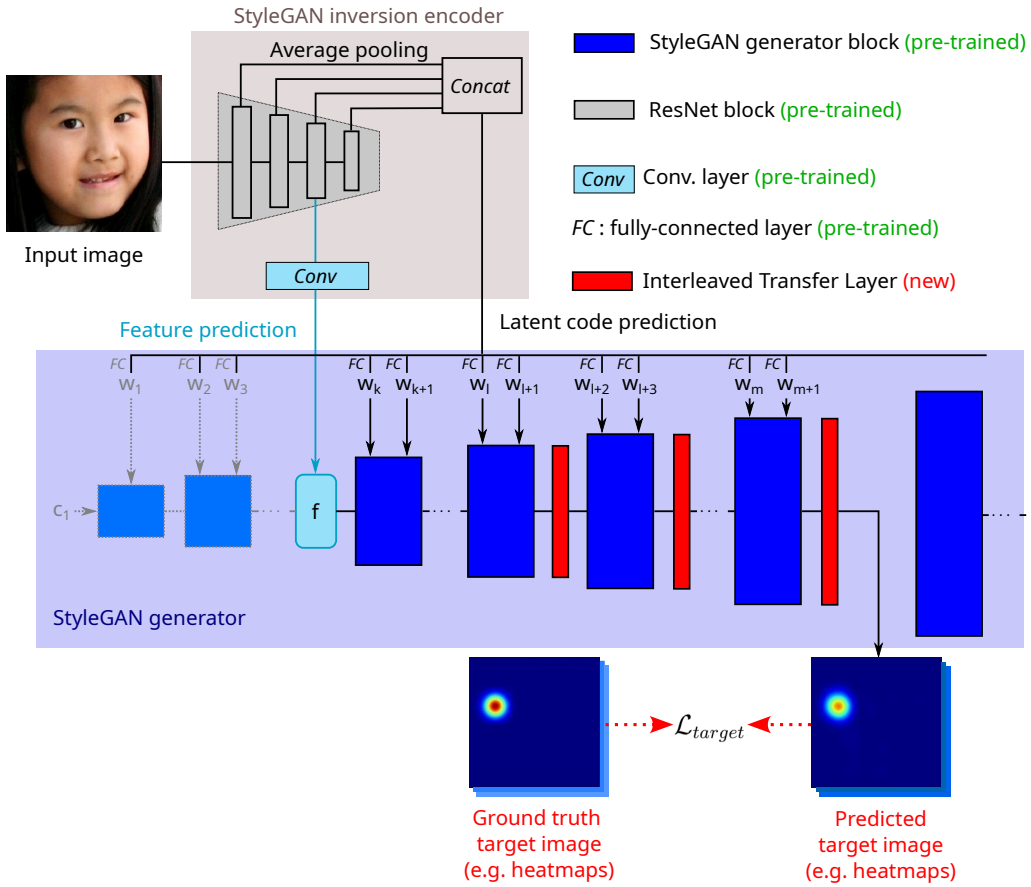


Figure 2.5 – GMDA-S version of the generative model modified for the supervised image-to-image translation task. Similar to the GMDA-R version, Interleaved Transfer Layers (ITLs) are added between the generator layers and are trained in a supervised manner to generate the target image of the supervised image-to-image translation task.

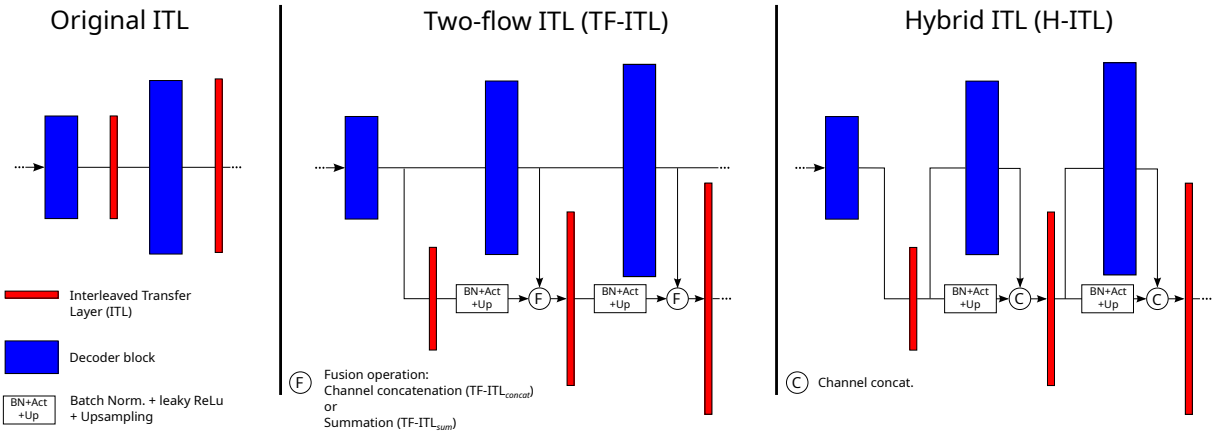


Figure 2.6 – The original ITL configuration [BW20] (left) and our two new configurations (middle and right).

To resolve these issues, we try two other ways to insert the ITLs in the decoder.

2.3.2 Two-flow Interleaved Transfer Layers (TF-ITL)

In this version, we add a direct flow between the ITLs. There are now 2 different flows: the face reconstruction flow through the decoder layers to generate the face image, which is this time remains untouched after the self-supervised training and this new "ITL flow" between the ITLs. Before each ITL, the original flow is merged with the ITL flow through either channel concatenation (TF-ITL$_{concat}$) or element-wise summation (TF-ITL$_{sum}$) depending on our experiment setup. Before this fusion operation, the ITL flow goes through Batch Normalization and Leaky ReLu operations and is upsampled to match the spatial dimensions of the original flow.

In this new version, the ITLs can still incorporate information from the decoder layers but they have more freedom to generate an optimal flow for the downstream task.

2.3.3 Hybrid Interleaved Transfer Layers (H-ITL)

We also test a hybrid approach. This version is a middle ground between the original version of the ITLs and the two-flow version. Like the two-flow version, there is a direct flow between the ITLs but like the original version, the flow through the decoder layers is changed (the face reconstruction flow). Basically, the output of an ITL is fed to both the next ITL and decoder layer. The output of the decoder layer is merged with the ITL flow using channel concatenation (we can't use summation because the number of output

channels of the previous ITL must be equal to the number of input channels of the decoder layer which has a different number of output channels).

The three different versions of ITLs are displayed in Figure 2.6.

2.4 Adding skip-connections (SC)

In the encoder, the spatial dimensions of the feature maps are gradually reduced as global information emerges in the encoder, resulting in a compact representation of the face image. While this representation contains valuable information that is useful for reconstructing the whole face, it may not contain sufficient local details for tasks that need precise detection. To resolve the issue, we take inspiration into the Hourglass Network [NYD16] by adding skip-connections between the encoder and the decoder prior to the supervised training, so that local details from the high-resolution encoder feature maps can improve the quality of the ITLs feature maps. In our modified architecture the input of an ITL is the element-wise sum of the output of the previous layer block of the decoder and the output of the corresponding encoder layer (the one with the same spatial dimensions) transformed by a set of convolution layers called "bottleneck block". For the two-flow and hybrid versions of the ITLs, the summation happens with the ITL flow before the fusion of the ITL flow and face reconstruction flow. Figure 2.7 displays how the skip-connections are added to our models depending on the ITL version.

2.5 Conclusion

In this Chapter, we have presented our methodology called Generative Model Decoder Adaptation (GMDA) to train a deep learning model with limited annotated training data for a supervised image-to-image translation task. We adapt a pre-trained generative model to the target task by re-using both the latent representation and the decoder layers for the target task. To re-use the decoder layer activations, we interleave additional convolution layers between the decoder layers and train them to generate the target image. By doing so, we can train the model with limited annotated data. We have presented two generative models that can be used: the GMDA-R version of our method uses an autoencoder based on ResNet as generative model and the GMDA-S versions uses StyleGAN. We have also proposed different ways to insert the additional convolutional layers. Finally, we have also proposed to add skip-connections between the encoder and the decoder layers to improve

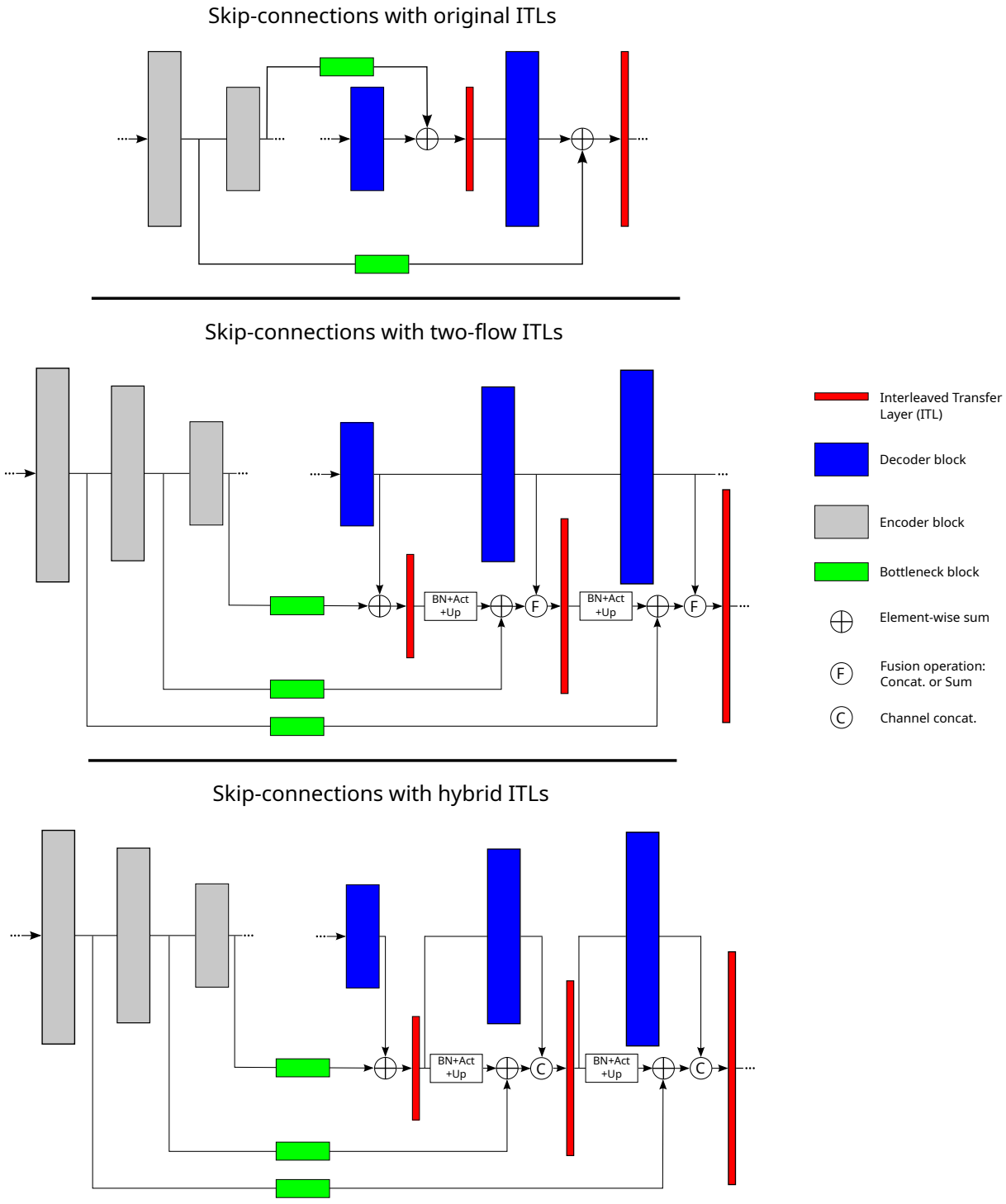


Figure 2.7 – Our models enhanced with skip-connections (SC) between the encoder and the ITLs. Top: skip-connections with the original ITLs. Middle: skip-connections with the two-flow ITLs. Bottom skip-connections with the hybrid ITLs. Skip-connections can be added to both GMDA-R or GMDA-S versions.

local details of the generated target image. In the following two chapters, we will present the results of experiments made with our methodology applied to two applications. Face alignment in Chapter 3 and 3D face reconstruction in Chapter 4.

Chapter 3

Application to face alignment

As explained in Section 1.4, annotating face images with landmark annotations is time-consuming. Also, because some landmarks, especially the ones on the outline of the face or occluded landmarks, are ambiguous, it introduces inconsistency between face images, particularly if there are multiple annotators. Most face alignment datasets suffer from these annotation errors which make the training of models with limited annotation even harder.

Having prior knowledge about what is a face should make a model easier to train, in terms of number of annotated samples needed, but also more robust to annotation errors. That is why we apply our Generative Model Decoder Adaptation (GMDA) methodology proposed in Chapter 2 to the face alignment task, in order to reduce the number of annotated samples needed to train a model, and increase its robustness. In this chapter we test the GMDA-R version (which uses a ResNet autoencoder as generative model) and the GMDA-S (which uses a StyleGAN autoencoder as generative model). We also test the different versions of the Interleaved Transfer Layers described in Section 2.3: the original version and our proposed two-flow (TF-ITL) and hybrid (H-ITL) versions. For the TF-ITL version, we test the two possible fusion operations between the ITL flow and reconstruction flow (refer to Section 2.3 for more details). The TF-ITL$_{concat}$ version uses channel concatenation as fusion operation and the TF-ITL$_{sum}$ version uses element-wise summation. Finally, we also test if the addition of skip-connections (SC) between the encoder and the decoder improves the model performance. This gives 16 settings to test. We also present in this chapter an active learning scheme dedicated to the face alignment task with limited training data

3.1 Application specificities

3.1.1 Adapting GMDA to face alignment

In Chapter 2, we detailed our GMDA methodology to train a model with limited annotated data. This principle involves adapting a pre-trained generative model, especially the decoder, to train it on a supervised image-to-image translation task using only a few annotated data.

As explained in Section 1.4.1, the face alignment task can be transformed into a supervised image-to-image translation task by making the network predict landmarks heatmaps instead of landmark positions. Thus, using a pre-trained generative model, we modify its decoder with the addition of Interleaved Transfer Layers [BW20]. Using landmarks annotations (transformed into landmark heatmaps), we train the modified generative model to predict landmark heatmaps in a supervised manner. Since most of the network parameters are already pre-trained, the training can be done using only a few annotations.

3.1.2 Active learning for face alignment

As presented in Section 1.3, active learning can be used to select the best samples to annotate to maximize the performance of the model even though it is trained on a small annotated dataset. These samples are selected using an acquisition function. We propose a new function called Negative Neighborhood Magnitude (NNM), based on uncertainty sampling, to assess the quality of the predicted heatmaps. During our experiments we have noticed that when the model is not confident, the magnitude of predicted heatmap near the predicted landmark location is low (see Figure 3.1). Adding these samples with low magnitude to the training set should increase the model performance. The NNM is designed to select this kind of samples, it is also designed to be fast to compute. To do so, we don't analyze the magnitude of the whole heatmap but only around the predicted landmark location (the peak of the heatmap). The computation of the NNM is as follows: First, for each predicted heatmap $\tilde{H}$, we compute the sum of the heatmap pixels within a square window W_i of size s centered around the predicted landmark position $\tilde{l}_i$. Next, we sum up all these sums and take the negative of the total sum. By doing this, the NNM's behavior becomes analogous to entropy, meaning that the higher the NNM value, the less

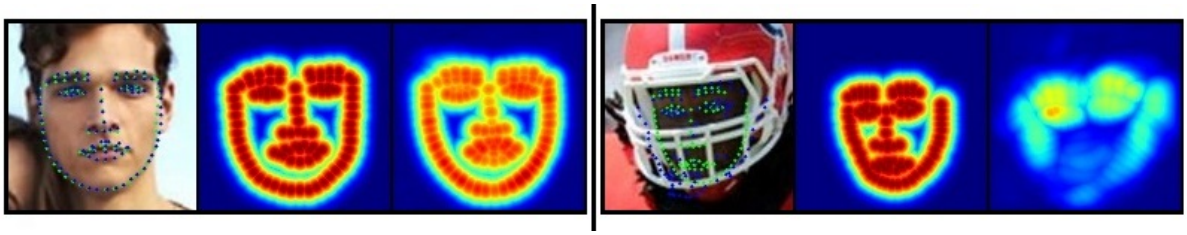


Figure 3.1 – Original image with ground truth (green dots) and predicted landmarks (blue dots), ground truth heatmaps and predicted heatmaps. Wrong landmarks predictions are usually associated with low magnitude heatmaps. Adding these samples to the training set should improve the model performance. Our proposed acquisition function, the Negative Neighborhood Magnitude (NNM), is designed to select this kind of samples.

confident the model is in its predictions, similar to how uncertainty increases with higher entropy. Thus, the formula is:

$$\mathrm{NNM}(\tilde{H}) = -\sum_{i=1}^{L} \sum_{u,v \in W_i} \tilde{H}_i(u,v) \ . \tag{3.1}$$

3.2 Datasets

To prove the effectiveness of our method and compare it to existing methods, we train and test our method on several face alignment datasets for both 2D and 3D facial landmarks.

3.2.1 Datasets for 2D face alignment

300-W. The 300-W dataset [Sag+13] is the aggregation of multiple 2D facial landmark datasets that have been re-annotated with 68 landmarks. The dataset is divided into 2 sets: a training set of 3148 face images and a *Full* test set of 689 face images. The Full test set is further divided into a *Common* test set of 554 face images where detecting the landmarks is fairly easy, and a *Challenging* test set of 135 with more challenging face images (occlusion, low-resolution...).

AFLW. This dataset [Koe+11] contains 24,386 face images annotated with 21 2D landmarks. Following usual practice [DY19; BW20], we ignore the landmarks of the ears and use 20,000 training images and 4,386 testing images. We evaluate our models on the

Full test set and the *Frontal* test set, a subset that contains only face images with a frontal view.

WFLW. This challenging dataset presented in Wu et al.'s publication [Wu+18] comprises a total of 10,000 images, including 7,500 training images and 2,500 testing images. Each image is annotated with 98 2D landmarks. The testing set is further divided into multiple subsets, which partially overlap with each other. Each subset emphasizes a specific characteristic, such as pose, expression, illumination, make-up, occlusion, or blur.

3.2.2 Datasets for 3D face alignment

300-W-LP: This is a synthetic dataset [Zhu+16] created from 300-W images using the profiling method of [Zhu+16] to render its faces into larger poses. This dataset contains 122,450 face images with a face pose yaw angle ranging from -90° to 90°. 68 2D and 3D landmarks annotations are provided for each face. We train our models on this dataset to predict the 3D landmarks.

AFLW2000-3D. This dataset [Zhu+16] was constructed by re-annotating the first 2,000 images of AFLW with 68 3D landmarks consistent with the ones of 300-W-LP. The face pose also ranges from -90° to 90°. We use this dataset to evaluate our models trained on 300-W-LP. The dataset can be divided into 3 subsets according to the absolute face pose (yaw angle): a subset with almost frontal faces ([0°, 30°]), another one with medium poses ([30°, 60°]) and the last one with profile views ([60°, 90°]).

While these two datasets are widely used, they have been annotated using a semi-automatic process [Zhu+16] so the quality of the annotations is poor on many images, as noticed by Bulat et al. [BT17b].

3.3 Experimental settings

3.3.1 Model architectures

Our models generate landmark heatmaps. Input images are resized to 256×256. We use 5 ITLs and output heatmaps of size 128×128 (3Fabrec [BW20] showed that the last convolution layer of the generator contains almost no face shape information). Landmark positions are computed as the argmax of each corresponding heatmap. The type of ITL used; original, our proposed two-flow (TF-ITL) or our proposed hybrid (H-ITL); depends on the experiment setting.

The GMDA-R version of our models uses the ResNet autoencoder proposed by 3FabRec [BW20] as generative model. The encoder is a ResNet18 and the decoder is an inverted ResNet18 using deconvolutions instead of convolutions. . We don't perform the self-supervised training ourselves but re-use pre-trained weights provided by the authors at the address `https://github.com/browatbn2/3FabRec`. The latent code dimension is 99 (3Fabrec authors' choice). For the skip-connections convolution blocks we use the hierarchical, parallel, and multi-scale block from [BT17a].

The GMDA-S version of our models use as decoder a StyleGAN2 [Kar+20] generator pre-trained on the FFHQ dataset [KLA19]. The encoder is a Feature-Style encoder [Yao+22] pre-trained on the same dataset.

3.3.2 Training

We use a traditional Mean Squared Error loss between the predicted and ground truth heatmaps as training loss. We train our models with Adam optimizer [KB15] using a batch size of 8. The ITLs and the pre-trained encoder are trained simultaneously.

For the GMDA-R version, the learning rate for the ITLs is 0.0004 and the learning rate for the pre-trained encoder is 0.00002. The number of epochs depends on the dataset and training set size. Except for Tables 3.7 and 3.8, the formula for the number of epochs is:

$$300 \times \sqrt{\frac{T_{full}}{T}} \quad , \tag{3.2}$$

where T_{full} is the total number of samples of the full training set and T is the number of samples from this training set used for training.

This formula is empiric and was used to reduce the number of cross-validation experiments to run in order to find the optimal number of epochs since we had many settings to test.

For the the GMDA-S version, the learning rate for the ITLs is 0.0001 and the learning rate for the pre-trained encoder is 0.00002. For all the results, except the ones from Table 3.7 and Table 3.8, the models are trained for 200,000 steps and both learning rates are decreased by a factor 0.995 every 10 epochs.

When training with active learning, the initial training set size is 10. The number of annotated samples added to the training set after a training depends on the final training set size, we make it large enough so that there is no more than 5 trainings in total. We use

the Negative Neighborhood Magnitude as acquisition function. We discard the top-10% ranked samples to avoid selecting outliers (this choice is explained in Section 3.4.2).

We use random vertical flip, rotation, translation, scaling, occlusions, Gaussian blur, brightness and contrast changes as data augmentations.

3.3.3 Evaluation

To evaluate our models, we use as main metric the commonly used Normalized Mean Error (NME). We also use the Area Under Curve (AUC) of the Cumulative Error Distribution (CED) and the Failure Rate (FR). The NME is defined as:

$$\text{NME } (\%) = \frac{1}{N} \sum_{i=1}^{N} \frac{||s_i - \tilde{s}_i||}{d} * 100 \ , \tag{3.3}$$

where s_i and $\tilde{s}_i$ are the ground truth and predicted location of landmark i, N the number of landmarks and d a normalization distance.

For 300-W and WFLW, we use the distance between the outer eye corners as the normalization distance for the NME ($\text{NME}_{\text{inter-ocular}}$). We report the FR and AUC at 10% NME ($\text{FR}^{10}_{\text{inter-ocular}}$, $\text{AUC}^{10}_{\text{inter-ocular}}$). For AFLW, because of the large number of profile faces, we report both the NME normalized with the diagonal of the ground truth bounding box (NME_{diag}) and the square root of the ground truth bounding box area (NME_{box}). We also report the AUC at 7% NME_{box} ($\text{AUC}^{7}_{\text{box}}$). For AFLW2000-3D, we also use NME_{box}. Because no bounding box is provided for this dataset, it is computed from the ground truth 3D landmarks.

When training with limited data (without active learning), we average the results of 5 five runs with random training samples (potentially overlapping). However, to obtain a better comparison between our different models, these 5 random training sets are the same across all models.

3.4 Results on 2D face alignment

We have mainly tested GMDA-R and GMDA-S architectures with the original ITLs (two-flow and hybrid versions were invented later during the PhD) so we have more results to provide with these two architectures. The skip-connections in the GMDA-S architecture were also introduced later. Also, some initial architecture results published were later refined through better hyper-parameter tuning. That is why results for the

same architecture might differ across different tables. When this happens, we explain the discrepancy.

3.4.1 Comparison with state-of-the-art

3.4.1.1 Fully-supervised

Although it is not our primary goal, we compare our models to fully-supervised face alignment when training on the whole training dataset.

Table 3.1 reports our results for the 300-W and WFLW datasets. We have tested our GMDA-R and GMDA-S versions with the original ITLs plus the enhanced GMDA-R version with skip-connections (SC) (also with original ITLs). The GMDA-R version without skip-connections shares the same architecture as 3FabRec [BW20] but it is our implementation with different learning size and training epochs.

For this version we obtain worse results compared to the ones of 3FabRec on 300-W but slightly better on WFLW. The addition of the skip-connections improves the results on both 300-W and WFLW but the most noticeable gap happens when we switch to the GMDA-S architecture with a reduction of 15% of the NME on the 300-W Full test set and 17% on the WFLW Full test set.

While our models don't obtain results as good as recent fully-supervised methods we are more interested in their results when training limited annotated data (presented later in this section).

The results on AFLW are presented in Table 3.2. Again the addition of skip-connections to the GMDA-R architecture improves the performance but the GMDA-S version still obtain the best results among the 3 configurations. This time, our models are closer to the state-of-the-art, GMDA-R with skip-connections and GMDA-S are only surpassed by FaRL [Zhe+22], another approach based on transfer learning but which does not use a generative model (refer to Section 1.4.4 for more details).

3.4.1.2 Semi-supervised

We also compare our models to other methods trained with limited annotated data. On 300-W (Table 3.3), while our GMDA-R obtains worse results than 3FabRec [BW20] when training on the whole training dataset, our model is better on small training datasets. The addition of the skip-connections to the GMDA-R architecture improves the NME except for training set size of 50 although this last result will be nuanced later (see Table 3.7).

	300-W			**WFLW**
Method	Com.	Chal.	Full	Full
LAB [Wu+18]	2.98	5.19	3.49	5.27
AVS [Qia+19]	3.21	6.49	3.86	4.39
AWing [WBF19]	2.72	4.52	3.07	4.36
LUVLi [Kum+20]	2.76	5.16	3.23	4.37
SHR-FAN [BST21]	2.61	**4.13**	2.94	**3.72**
ADNet [Hua+21]	**2.53**	4.58	**2.93**	4.14
FaRL [Zhe+22]	2.56	4.45	**2.93**	3.96
Wood et al.* [Woo+22]	3.03	4.80	3.38	-
VGG-F [Bul+22]	-	-	3.20	4.57
3FabRec [BW20]	3.36	5.74	3.82	5.62
GMDA-R** (Ours)	3.54	5.93	4.01	5.58
GMDA-R+SC (Ours)	3.48	5.83	3.95	5.50
GMDA-S (Ours)	2.97	5.30	3.42	4.62

Table 3.1 – Comparison with state-of-the-art face alignment methods when training with the whole training set of 300-W or WFLW (except for Wood et al.). The scores displayed are the $\text{NME}_{\text{inter-ocular}}$ on the 300-W Common, Challenging and Full test sets, and on the WFLW Full test set. Compared to fully-supervised methods, our models, designed to be trained with limited annotated data, fall behind when the training data is abundant. GMDA-S performs better than GMDA-R. *Trained on a synthetic dataset. **Same architecture as 3FabRec but our implementation. "SC" is skip-connections.

AFLW dataset				
	$\text{NME}_{\text{diag}} \downarrow$		$\text{NME}_{\text{box}} \downarrow$	$\text{AUC}^{7}_{\text{box}} \uparrow$
Method	Full	Frontal	Full	Full
LAB [Wu+18]	1.25	1.14	-	-
HR-Net [Sun+19]	1.57	1.46	-	-
LUVLi [Kum+20]	1.39	1.19	2.28	0.680
3FabRec [BW20]	-	-	1.84	-
SHR-FAN [BST21]	1.31	1.19	2.14	0.700
VGG-F [Bul+22]	1.54	-	-	-
FaRL [Zhe+22]	**0.94**	**0.82**	**1.33**	**0.813**
GMDA-R (Ours)	1.28	1.09	1.81	0.740
GMDA-R+SC (Ours)	1.19	1.03	1.68	0.759
GMDA-S (Ours)	1.02	0.90	1.45	0.791

Table 3.2 – Comparison with state-of-the-art methods on the AFLW Full and Frontal test sets when training on the whole AFLW training set. Our models are better than many fully-supervised methods. Again, GMDA-S performs better than GMDA-R.

300-W dataset															
Method	Training set size														
	3148 (100%)			630 (20%)			315 (10%)			168 (5%)			50 (1.59%)		
	Com.	Ch.	Full	Com.	Ch.	Full	Com.	Ch.	Full	Com.	Ch.	Full	Com.	Ch.	Full
RCN+ [Hon+18]	3.00	**4.98**	3.46	-	6.12	4.15	-	6.63	4.47	-	9.95	5.11	-	-	-
AVS [Qia+19]	3.21	6.49	3.86	3.85	-	-	4.27	-	-	6.32	-	-	-	-	-
TS^3 [DY19]	**2.91**	5.90	3.49	4.31	7.97	5.03	4.67	9.26	5.64	-	-	-	-	-	-
3FabRec [BW20]	3.36	5.74	3.82	3.76	6.53	4.31	3.88	6.88	4.47	4.22	6.95	4.75	4.55	7.39	5.10
VGG-F* [Bul+22]	-	-	**3.20**	-	-	-	-	-	**3.48**	-	-	-	-	-	4.13
GMDA-R (Ours)	3.54	5.93	4.01	3.79	6.33	4.29	3.93	6.70	4.47	4.10	6.86	4.64	4.27	7.23	4.85
GMDA-R+SC (Ours)	3.48	5.89	3.95	3.66	6.23	4.17	3.87	6.60	4.40	3.93	6.84	4.50	4.33	7.60	4.97
GMDA-R+SC+AL (Ours)	-	-	-	-	-	-	3.99	6.49	4.48	4.19	6.78	4.70	4.29	6.93	4.81
GMDA-S (Ours)	2.97	5.30	3.42	3.14	5.66	3.64	3.22	5.87	3.74	3.33	6.05	3.86	3.57	6.62	4.16
GMDA-S+AL (Ours)	-	-	-	**3.12**	**5.53**	**3.59**	**3.20**	**5.67**	3.68	**3.32**	**5.83**	**3.81**	**3.54**	**6.24**	**4.06**

Table 3.3 – Comparison ($NME_{\text{inter-ocular}}$) with other semi-supervised methods on the 300-W on the Common, Challenging and Full test sets when training with limited annotated data. "AL" is Active Learning, "SC" is skip-connections. Bold is best, underlined is second best. GMDA-S is better than GMDA-R and almost all existing methods. *Hyper-parameters fine-tuned for each training size.

Switching to the GMDA-S architecture greatly improves the NME, even on small training set sizes meaning that the StyleGAN encoder can be correctly fine-tuned with only a few annotated samples.

Apart from VGG-F [Bul+22], our GMDA-S model outperforms other semi-supervised methods. As explained in their paper, VGG-F fined-tuned the learning rate, learning rate scheduler and number of epochs for each dataset and training set size which led to an extraordinary number of experiments to run as the admit. In our case, because we didn't want to spend too much time on hyper-parameter fine-tuning, our GMDA-S models used, apart for the number of epochs, the same hyper-parameters (see Section 3.3.2). When using active learning, we surpass VGG-F on the 50 training samples setting.

Results for WFLW are reported in Table 3.4. Like 300-W results, the use of skip-connections improves the NME except when training with 50 examples (this result will be nuanced later in Table 3.7). The GMDA-S obtains better results than GMDA-R except when training with 50 samples. This may be explained by the fact that the images of WFLW are sometimes very different in terms of pose or resolution compared to the ones found in the FFHQ dataset (80K images) that has been used to train the StyleGAN encoder and generator of GMDA-S. In comparison, the ResNet autoencoder of GMDA-R has been trained on more varied images (2M images). Thus, GMDA-S needs more annotated samples to adapt its predictions.

For AFLW (Table 3.5) , we only test our GMDA-S because it was the one which obtains the best results on 300-W and WFLW. Since existing methods present their

WFLW dataset					
Method	Training set size				
	7500 (100%)	1500 (20%)	750 (10%)	375 (5%)	50 (0.67%)
AVS [Qia+19]	**4.39**	6.00	7.20	-	-
3FabRec [BW20]	5.62	6.51	6.73	7.68	8.39
VGG-F* [Bul+22]	<u>4.57</u>	-	<u>5.44</u>	-	**7.11**
GMDA-R (Ours)	5.58	6.23	6.42	6.84	7.74
GMDA-R+SC (Ours)	5.50	6.07	6.28	6.72	8.06
GMDA-R+SC+AL (Ours)	-	-	6.24	6.59	7.60
GMDA-S (Ours)	4.62	<u>5.09</u>	<u>5.44</u>	<u>5.80</u>	7.78
GMDA-S+AL (Ours)	-	**4.94**	**5.18**	**5.45**	<u>7.30</u>

Table 3.4 – Comparison ($NME_{\text{inter-ocular}}$) with other semi-supervised methods on the WFLW Full test set when training with limited annotated data. "AL" is Active Learning. Bold is best, underlined is second best. GMDA-S performs better than GMDA-R. *Hyper-parameters fine-tuned for each training size.

results using NME_{diag} or NME_{box}, we compute both metrics. Regarding NME_{diag}, our approach outperforms other semi-supervised methods across all training set sizes for both the Full and Frontal test sets. For NME_{box}, we compare our method against two other semi-supervised approaches, FaRL [Zhe+22] and VGG-F [Bul+22]. We surpass VGG-F on all training sizes even though they performed heavy hyper-parameter fine-tuning. As for the comparison with FaRL, although they perform better when training on the full training set, we obtain similar results for the 10% training size and surpass their performance when training with only 1% of the training dataset.

3.4.2 Training with active learning

3.4.2.1 Selecting the acquisition function

To study the effectiveness of our proposed Negative Neighborhood Magnitude (NNM), we have conducted experiments using three distinct acquisition functions for the GMDA-R architecture with and without skip-connections. Two of these functions relies on uncertainty sampling: the NNM and the mean of the spatial entropy of the heatmaps. The third function, based on diversity sampling, is K-center-greedy algorithm described in [SS18].

Table 3.6 presents the $NME_{\text{inter-ocular}}$ results on WFLW for the different acquisition functions. We chose this dataset for his numerous number of challenging images in both training and test sets which makes it a good candidate to test the efficiency of active

AFLW dataset												
Method	Training set size											
	20000 (100%)		4000 (20%)		2000 (10%)		1000 (5%)		200 (1%)		50 (0.25%)	
	Full	Fr.	Full	Fr.	Full	Fr.	Full	Fr.	Full	Fr.	Full	Fr.
NME_{box}												
RCN+ [Hon+18]	1.61	-	-	-	-	-	2.17	-	2.88	-	-	-
TS^3 [DY19]	-	-	1.99	1.86	2.14	1.94	2.19	2.03	-	-	-	-
3FabRec [BW20]	1.87	1.59	1.96	1.74	2.03	1.74	2.13	1.86	2.38	2.03	2.74	2.23
GMDA-S	**1.45**	**1.28**	**1.60**	**1.39**	**1.63**	**1.41**	**1.66**	**1.43**	1.79	**1.53**	**2.05**	**1.71**
GMDA-S+AL	-	-	-	-	-	-	**1.66**	1.49	**1.77**	1.56	**2.03**	**1.75**
NME_{diag}												
FaRL [Zhe+22]	**0.94**	**0.82**	-	-	**1.15**	-	-	-	1.35	-	-	-
VGG-F* [Bul+22]	1.54	-	-	-	1.70	-	-	-	1.91	-	-	-
GMDA-S	1.02	0.90	**1.13**	**0.98**	**1.15**	**1.00**	**1.17**	**1.01**	1.27	**1.08**	**1.45**	**1.21**
GMDA-S+AL	-	-	-	-	-	-	**1.17**	1.05	**1.25**	1.11	**1.44**	**1.24**

Table 3.5 – Comparison with other semi-supervised methods on AFLW on the Full and Frontal (Fr.) test sets when training with limited annotated data. Our GMDA-S models outperform all other methods when training with small training set sizes.*Hyper-parameters fine-tuned for each training size.

learning. Except for GMDA-R without skip-connections with a final training size of 50, the K-center-greedy method consistently outperforms random sampling. On the other hand, when using NNM or Entropy to select samples from *all* the unlabeled data and when the final training size is small ($\leq$200), the results are either worse or only marginally better than random sampling.

Nonetheless, a significant improvement can be achieved by excluding the top 10% ranked samples during the sample selection process, leading to better NME scores compared to random and K-center-greedy sampling. These improved methods are referred to as $NNM_{10\%}$ and $Entropy_{10\%}$ in Table 3.6. This finding indicates that very challenging samples in the WFLW training dataset are actually outliers and should be avoided since their inclusion does not contribute to the model's generalization to unseen data.

As the final training set size increases, the advantage of excluding the worst examples tends to diminish. This is because with the addition of more samples, the proportion of outliers decreases, and a more representative set of "normal" challenging samples is introduced into the training set. Thus, their inclusion becomes beneficial for the model's performance.

Fig. 3.2 illustrates the top-5 ranked samples based on the NNM (Negative Neighborhood Magnitude) after training our model on 10 randomly selected samples from the

Acquisition function comparison										
Acquisition function	GMDA-R					GMDA-R+SC				
	Final training set size					Final training set size				
	50	100	200	5%	10%	50	100	200	5%	10%
Random	7.74	7.44	7.04	6.84	6.42	8.06	7.40	6.88	6.72	6.28
NNM	8.27	7.57	7.15	6.77	6.36	8.04	7.44	7.01	6.63	6.22
Entropy	8.17	7.53	7.06	6.71	6.32	7.95	7.44	7.02	6.61	6.22
$\text{NNM}_{10\%}$	**7.63**	7.20	6.82	**6.62**	**6.31**	7.60	6.99	**6.72**	**6.59**	6.24
$\text{Entropy}_{10\%}$	7.71	**7.12**	**6.83**	**6.62**	6.34	**7.53**	**6.96**	6.73	6.62	**6.22**
K-center-greedy	7.85	7.36	6.95	6.65	6.32	7.74	7.18	6.82	6.61	6.28

Table 3.6 – $\text{NME}_{\text{inter-ocular}}$ on WFLW Full test set for different active learning methods and different training set sizes (5% = 375 examples and 10% = 750 examples), for our GMDA-R architectures. $\text{Entropy}_{10\%}$ and our proposed $\text{NNM}_{10\%}$ have the best results by excluding the top-10% ranked samples to avoid selecting outliers while still selecting challenging examples.

WFLW training set. In the top row, we observe the top-5 samples selected from *all* the unlabeled samples. These five images stand out as outliers: blue faces, the second image displays a distorted face, and the last image is of a non-human face. Including such outlier images in the training set is unlikely to significantly benefit the model's generalization to unseen data. On the other hand, in the bottom row, we see the top-5 images after removing the top-10% ranked images from the unlabeled dataset. Despite their challenging nature, such as low resolution, occlusion, and baby faces, these five images are more "normal" images compared to ones of the top row. Consequently, the model is likely to gain better predictive capabilities if these samples are added to the training dataset.

Entropy and NNM demonstrate close NME results in our experiments. However, there is a noteworthy difference in their computation requirements. For Entropy, the entire heatmaps must be normalized before computing the entropy, while NNM only requires summing heatmap values within small windows. As a consequence, computing the Entropy on average took approximately 0.042 seconds using an Intel Core i7-9850H CPU, whereas computing the NNM only required 0.012 seconds. Therefore, the NNM is approximately 3.5 times faster to compute compared to Entropy, while still achieving comparable performance results in terms of NME. This computational advantage makes NNM a more efficient choice for sample selection, which can be especially beneficial when dealing with large datasets.

Figure 3.2 – Top-5 ranked images for NNM after training the model on 10 random samples of WFLW. Ground truth landmarks are displayed with green dots while blue ones are the predicted landmarks. Top row shows the top-5 ranked images among all the unlabeled samples while bottom row displays the top-5 ranked images after removing the top-10% images. Top images are clearly outliers while bottom images are more natural but still challenging images (low-resolution, occlusion,..).

3.4.2.2 Results with active learning

For 300-W (see Table 3.3), when using Active Learning to select the training samples, results differ between GMDA-R and GMDA-S models. For GMDA-R, the NME decreases on the Challenging test set which contains difficult face images (up to a 9% decrease for the training size of 50) while it increases for the Common test set with easy face images (less than a 7% NME increase at most). This means that the active learning procedure samples mainly difficult images thus the model predictions improves on this kind of images but with the cost of a slightly reduced accuracy on easy faces. In the case of the GMDA-S, the NME is improved on the Challenging test set but also slightly on the Common test set. This suggests that the model does not need many easy training faces to predict correctly this kind of faces and benefits from having more hard faces in the training set. One could notice that GMDA-S trained with active learning using 5% of 300-W training dataset has a better NME on the Challenging test set compared to GMDA-S trained without active learning using 10% of 300-W training dataset. Thus, thanks to active learning, it makes it possible, in some cases, to halve the number of training samples to obtain similar or even better performance.

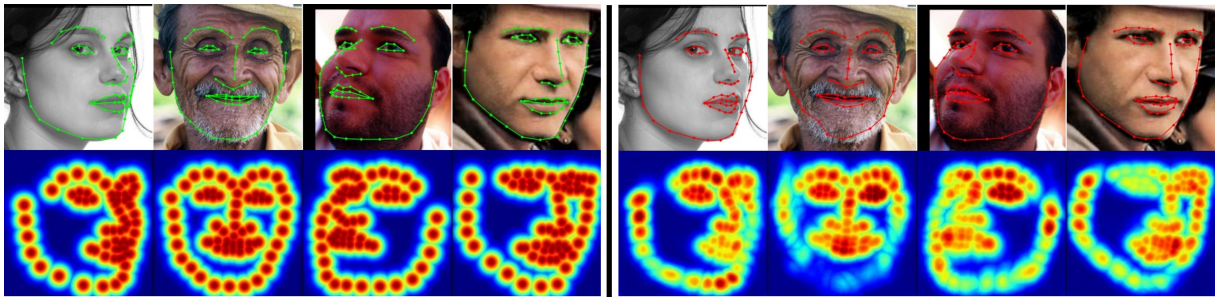


Figure 3.3 – Images of 300-W selected by active learning, after an initial training on 10 random images. Top left: ground truth (GT) landmarks. Top right: predicted landmarks. Bottom left: GT heatmaps. Bottom right: predicted heatmaps. The active learning acquisition function successfully detects inaccurate predictions.

In the case of WFLW (see Table 3.4) using active learning always improves the performance of both GMDA-R and GMDA-S architectures (up to a 6% NME decrease). For AFLW (Table 3.5), training with active learning does not lead to substantial performance improvements on the Full test set. However, it does result in a slight decrease in performance on the Frontal test set. This indicates that active learning enhances the model's performance on challenging images, particularly those with profile faces, but comes at the expense of a slightly reduced accuracy when dealing with frontal faces.

Figure 3.3 displays examples of images selected using the Negative Neighborhood Magnitude (NNM), the active learning acquisition function. Heatmaps with low magnitude can be attributed to several factors, such as network confusion between the outline of the face and hair or beard, unusual face poses, or nearly closed eyes. Moreover, landmarks located on the outline of the face tend to be the most ambiguous, resulting in their heatmap values being the lowest among all landmarks.

3.4.3 Architecture selection

To evaluate our proposed versions of the Interleaved Transfer Layers (ITLs) (see Chapter 3, Section 2.3) we have conducted multiple experiments on WFLW with different model architectures and training set sizes. Our results are reported in Table 3.7. We also test how skip-connections perform in the StyleGAN architecture. The training hyper-parameters are same as the ones presented in Section 3.3.2 except for number of epochs. We fine-tune it using 5-fold-cross-validation on the first samples of the training set. For example, when training with only 50 samples, we use the first 50 samples of the dataset for the

Model selection results						
Method	Training set size					
	50 (0.67%)		375 (5%)		750 (10%)	
	Mean NME	Med. NME	Mean NME	Med. NME	Mean NME	Med. NME
GMDA-R	7.61 ± 0.14	5.95 ± 0.10	6.47 ± 0.09	5.23 ± 0.03	6.18 ± 0.06	5.05 ± 0.07
GMDA-R w/ TF-ITL$_{sum}$	7.57 ± 0.19	5.87 ± 0.09	6.39 ± 0.09	5.17 ± 0.03	6.15 ± 0.06	5.02 ± 0.06
GMDA-R w/ TF-ITL$_{concat}$	7.62 ± 0.12	5.90 ± 0.08	6.43 ± 0.12	5.20 ± 0.05	6.14 ± 0.05	5.03 ± 0.06
GMDA-R w/ H-ITL	7.57 ± 0.16	5.92 ± 0.08	6.48 ± 0.12	5.24 ± 0.04	6.13 ± 0.06	5.01 ± 0.04
GMDA-R + SC	7.36 ± 0.08	5.75 ± 0.08	6.18 ± 0.10	5.02 ± 0.04	5.93 ± 0.09	4.85 ± 0.09
GMDA-R w/ TF-ITL$_{sum}$ + SC	7.36 ± 0.09	5.74 ± 0.16	6.17 ± 0.10	4.98 ± 0.02	5.84 ± 0.04	4.78 ± 0.04
GMDA-R w/ TF-ITL$_{concat}$ + SC	7.26 ± 0.17	5.62 ± 0.14	6.06 ± 0.07	4.90 ± 0.04	5.79 ± 0.04	4.70 ± 0.01
GMDA-R w/ H-ITL + SC	7.15 ± 0.11	5.57 ± 0.10	6.04 ± 0.09	4.90 ± 0.04	5.78 ± 0.04	4.71 ± 0.03
GMDA-S	7.78 ± 0.20	5.32 ± 0.09	5.93 ± 0.15	4.46 ± 0.03	5.52 ± 0.04	4.24 ± 0.03
GMDA-S w/ TF-ITL$_{sum}$	7.34 ± 0.24	5.35 ± 0.07	5.91 ± 0.08	4.46 ± 0.02	5.54 ± 0.06	4.26 ± 0.01
GMDA-S w/ TF-ITL$_{concat}$	7.43 ± 0.24	5.33 ± 0.07	5.90 ± 0.08	4.46 ± 0.01	5.54 ± 0.06	4.26 ± 0.03
GMDA-S w/ H-ITL	7.41 ± 0.08	5.34 ± 0.08	5.99 ± 0.08	4.46 ± 0.02	5.53 ± 0.09	4.25 ± 0.05
GMDA-S + SC	8.25 ± 0.33	5.44 ± 0.10	5.87 ± 0.05	4.44 ± 0.04	5.44 ± 0.04	4.25 ± 0.01
GMDA-S w/ TF-ITL$_{sum}$ + SC	8.26 ± 0.28	5.44 ± 0.07	5.86 ± 0.07	4.47 ± 0.03	5.45 ± 0.05	4.26 ± 0.03
GMDA-S w/ TF-ITL$_{concat}$ + SC	8.28 ± 0.31	5.44 ± 0.08	5.89 ± 0.07	4.48 ± 0.03	5.48 ± 0.10	4.26 ± 0.02
GMDA-S w/ H-ITL + SC	8.13 ± 0.13	5.44 ± 0.07	5.96 ± 0.07	4.46 ± 0.01	5.58 ± 0.11	4.25 ± 0.02

Table 3.7 – Comparison of our different versions of our architecture when training with different training set sizes. Training samples are sampled from the WFLW training set. Mean and median $\text{NME}_{\text{inter-ocular}}$ on the WFLW Full test set are reported. "TF-ITL" stands for Two-flow ITL, "H-ITL" for Hybrid ITL and "SC" for skip-connections.

cross-validation. If we train with 100 samples, we would use the 100 first samples of the dataset for the cross-validation. We also use constant learning rates. These different hyper-parameters explain the discrepancy between the results presented in Table 3.7 compared to the ones from previous tables. With all these experiments we hope to find the optimal combination of model parameters.

The three factors of variation are the generative model, the use of skip-connections and the different versions of the ITLs.

3.4.3.1 Generative model: GMDA-R vs GMDA-S

According to Table 3.7 which reports results for models trained and tested on WFLW, for the small training size 50, the GMDA-R versions (with skip-connections) obtain better results in terms of Mean $\text{NME}_{\text{inter-ocular}}$ compared to the GMDA-S versions (with or without skip-connections). However, the GMDA-S versions have better Median $\text{NME}_{\text{inter-ocular}}$ meaning they usually predict more accurate landmarks but have large NME on some images. This assumption is corroborated when looking at the Failure Rate for several thresholds (see Figure 3.4). While the GMDA-S version (original ITL, no skip-connections in the figure) has slightly fewer images with a NME superior to 10, it has many more images with a NME superior to 20 or 30.

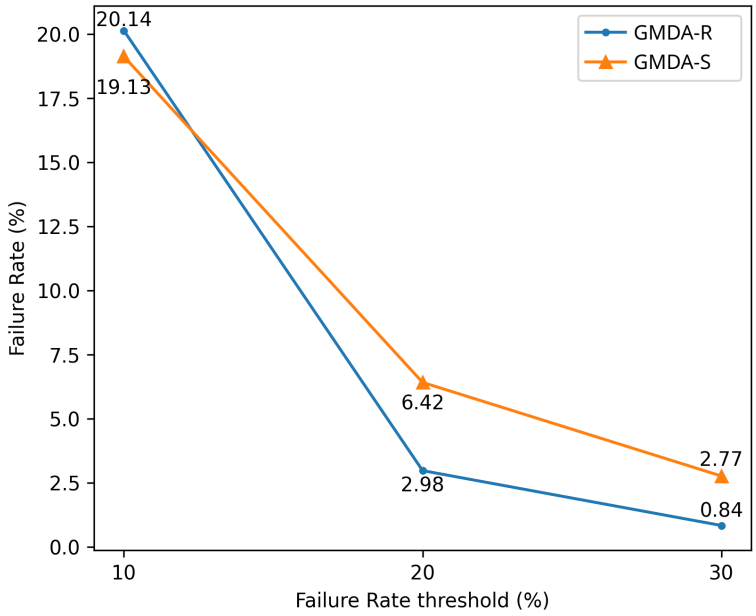


Figure 3.4 – Failure rate on WFLW Full test set depending on the failure rate threshold. Models have been trained on 50 samples. GMDA-S has more images with a very high NME than GMDA-R meaning that GMDA-S completely fails on some images but is generally more accurate than GMDA-R.

Figure 3.5 displays the predictions of GMDA-R and GMDA-S, trained with 50 images, on some test images of WFLW. We can see that GMDA-S predictions are sometimes completely off, especially for low-resolution images, but better for frontal and high-resolution images. This may be explained by the fact that low-resolution images are not present in FFHQ, the dataset used to train the generative model used by GMDA-S.

However, if we look at Table 3.8 which also reports results for models trained and tested on 300-W, a dataset which contains fewer challenging images compared to WFLW, this time GMDA-S (fine-tuned) has a better Mean NME than GMDA-R, even when training with only 50 samples. This last table does not test all the versions of the ITLs but focuses on encoder fine-tuning that we will discuss later.

When the training size increases, the GMDA-S versions obtain better Mean NME in addition to better Median NME since the model has a bigger capacity and can better adapt its representations to challenging face images.

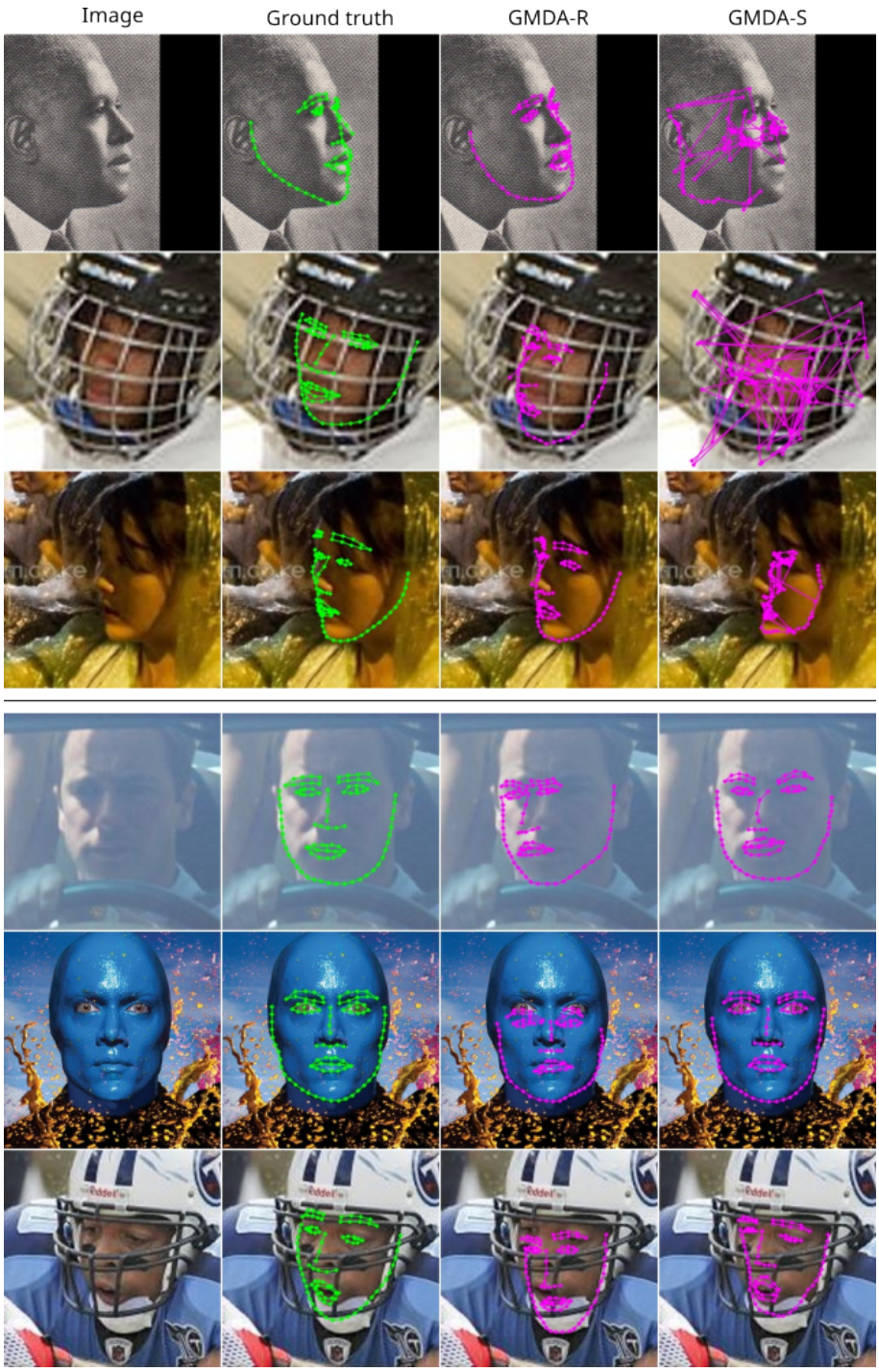


Figure 3.5 – Comparison of the landmarks predictions of GMDA-R and GMDA-S versions trained on 50 samples from WFLW. GMDA-S struggles on low resolution or profile images (top 3 images) but performs better on frontal and high resolution images (bottom 3 images). Green dots: ground truth landmarks, purple dots: predicted landmarks. For better visualization, the landmarks from a same semantic part of the face (mouth, eye, ...) are connected with lines.

3.4.3.2 Addition of skip-connections

Unlike the results presented in Tables 3.3 and 3.4, this time (Table 3.7), with our better hyper-parameters, the skip-connections in the GMDA-R architecture improve the results for all training sizes, even 50 samples. The mean NME decreases between 3% and 5%. However for the GMDA-S architecture, the skip-connections greatly increase the test NME when training with 50 samples. When the training size increases, the influence of skip-connections for the GMDA-S architecture fades away and it obtains similar results to the GMDA-S architecture without skip-connections. As conclusion, when training with limited data, skip-connections are useful in the GMDA-R architecture but not in the GMDA-S architecture. This might be due to StyleGAN unusual network flow (the latent code is added at multiple layers through Adaptive Normalization [HB17]). With additional work, it might be possible to find a better way to incorporate the skip-connections into the architecture.

3.4.3.3 Interleaved Transfer Layers

For the GMDA-R architecture without skip-connections, the different versions of the ITLs obtain similar results for every training set size. However, when skip-connections are added, the two-flow ITL version, with channel concatenation has fusion operation between the ITL and reconstruction flow (TF-ITL$_{concat}$), and even more the hybrid ITL version (Hyb. ITL) get better results with a 3% decrease of the mean NME in average. For the GMDA-S architecture, for the small training size of 50 the variances are quite high so it is hard to draw any conclusion but two-flow and hybrid versions seem to have a better Mean NME although the Median NME is the same. There is almost no difference between the different versions of the ITLs for the GMDA-S architecture when the training size is high (superior to 50). As conclusion, the hybrid ITL version gives the best results when combined with the GMDA-R architecture and with skip-connections. The gain is noticeable for all training sizes. Outside of this particular architecture, the different versions of ITLs give similar results.

3.4.3.4 Encoder fine-tuning

By default, we fine-tune the encoder while training the ITLs. However, when the training set size is too small, there is a risk that this fine-tuning might degrade the performance because of the increased number of parameters to optimize (although the encoder layers

	300-W			WFLW		
	Training set size			Training set size		
	3148 (100%)	315 (10%)	50 (1.59%)	7500 (100%)	750 (10%)	50 (0.67%)
GMDA-R w/o FT	4.27 ± 0.07	4.48 ± 0.07	5.24 ± 0.13	6.51 ± 0.04	6.92 ± 0.03	8.42 ± 0.16
GMDA-R w/ FT	4.01 ± 0.07	4.45 ± 0.08	4.95 ± 0.08	5.44 ± 0.08	6.18 ± 0.06	7.61 ± 0.14
GMDA-R+SC w/o FT	3.97 ± 0.05	4.31 ± 0.04	4.87 ± 0.11	5.65 ± 0.03	6.32 ± 0.08	7.82 ± 0.19
GMDA-R+SC w/ FT	3.90 ± 0.11	4.19 ± 0.02	4.72 ± 0.03	5.21 ± 0.02	5.93 ± 0.09	7.36 ± 0.08
GMDA-S w/o FT	4.54 ± 0.02	4.86 ± 0.03	5.99 ± 0.06	8.94 ± 0.08	9.42 ± 0.06	14.36 ± 0.33
GMDA-S w/ FT	3.42 ± 0.02	3.74 ± 0.03	4.16 ± 0.07	4.62 ± 0.03	5.52 ± 0.04	7.78 ± 0.20

Table 3.8 – Mean $\text{NME}_{\text{inter-ocular}}$ on 300-W and WFLW Full test sets for different training set sizes with and without encoder fine-tuning (FT). "SC" is skip-connections. Fine-tuning the encoder improves the NME in any case, especially for GMDA-S.

are already pre-trained and the learning rate is smaller for them). To check if this happens, we have experimented on 300-W and WFLW with three different architectures: the GMDA-R architecture, the GMDA-R architecture with skip-connections and the GMDA-S architecture. We use the original ITLs. The results of our experiments are reported in Table 3.8. For all three architectures, the fine-tuning of the encoder improves the model performance even when the training set size is very small (50 examples). For the GMDA-S architecture, without fine-tuning, the model performs poorly. We suppose that this is due to the fact that the images from the face alignment datasets, especially WFLW, may be very different from the ones found in FFHQ, the dataset on which the StyleGAN encoder and generator of the GMDA-S architecture have been trained. In this dataset, the face is aligned and the images are high-resolution. On the contrary, images from face alignment datasets are more varied (low resolution, occlusions, rotated face...). So, without fine-tuning, the StyleGAN representation struggles to encode the face in a way useful for the face alignment task although only a few annotated samples are needed for the fine-tuning.

3.5 Results on 3D face alignment

3.5.1 Comparison with fully-supervised methods

We have not found any existing method training on a reduced set of 300-W-LP so we compare our models to models which train on the whole dataset (we also train on the whole dataset for a fair comparison). We evaluate on the AFLW2000-3D dataset. Table 3.9 reports our results. If we compare our models with each other, the GMDA-S models obtains better results on frontal images (yaw angle inferior to 30°) compared to the GMDA-R models. However for profile images (yaw angle superior to 60°), the GMDA-

AFLW2000-3D dataset					
Method	0-30°	30-60°	60-90°	Balanced	Mean
3DDFA [Zhu+16]	3.78	4.54	7.93	5.42	6.03
3D-FAN [BT17b]	3.16	3.53	4.60	3.79	-
PRNet [Fen+18a]	2.75	3.51	4.61	3.62	3.26
3DDFAv2 [Guo+20]	**2.63**	3.42	4.48	3.51	-
SADRNet [Rua+21]	2.66	**3.30**	4.42	3.46	3.05
SynergyNet [WXN21]	2.65	**3.30**	**4.27**	**3.41**	-
GMDA-R (Ours)	2.77	3.57	4.74	3.69	3.22
GMDA-R+SC (Ours)	2.74	3.57	4.78	3.70	3.21
GMDA-S (Ours)	2.65	3.62	4.89	3.72	3.14

Table 3.9 – Comparison (NME_{box}) with *fully-supervised* methods on subsets of AFLW2000-3D divided by face pose (yaw angle). "Balanced" column is the average of the first 3 columns. "Mean" column reports the mean NME over the whole AFLW2000-3D dataset. All the models have been trained on the 300-W-LP dataset. Our models are close to state-of-the-art for profile faces but fall behind for larger face poses.

R models perform better. We suppose that because the self-supervised training of the ResNet autoencoder of the GMDA-R models was done on more varied images (2 millions images) compared to the self-supervised training of the StyleGAN of the GMDA-S models on FFHQ (80,000 images) which mainly contains frontal images, the GMDA-R models can better encode profile images which leads to more accurate landmark predictions. On the opposite, the GMDA-S reconstruction of frontal images is better so the landmark predictions are better for this kind of images. Compared to existing methods, our models obtains competitive results on frontal images but fall behind for images with a larger yaw face angle. This can also be explained that our model does not predict a 3D face model as most existing methods do but relies on landmark heatmaps which are not really well suited for large poses since many landmarks are occluded.

3.5.1.1 Results when training with limited data

We evaluate our model (only the GMDA-S version because of lack of time) on AFLW2000-3D after training on data sampled from 300-W-LP. We use three different sampling methods, "Random": fully random sampling, "Balanced": random sampling but the low (0-30°), medium (30-60°) and large pose (60-90°) subsets must have equal size and "Active": sampling using active learning (same procedure as used for 2D face alignment). We use different training set sizes, 300, 150, and 48, all divisible by 3 so we can have perfectly bal-

300-W-LP/AFLW2000-3D									
Sampling method	Training set size (300-W-LP)								
	300			150			48		
	0-30°	30-60°	60-90°	0-30°	30-60°	60-90°	0-30°	30-60°	60-90°
	Test NME_{box} (AFLW2000-3D)								
Random	2.77	**3.73**	5.00	2.90	**3.92**	5.25	3.43	4.73	6.08
std dev.	0.04	0.06	0.09	0.05	0.08	0.11	0.18	0.12	0.21
Balanced	**2.74**	3.76	5.05	**2.86**	3.93	5.25	**3.23**	**4.58**	6.35
std dev.	0.03	0.04	0.06	0.03	0.06	0.09	0.06	0.08	0.27
Active	2.81	3.75	**4.92**	2.98	3.94	**5.09**	3.45	4.70	**5.59**
std dev.	0.03	0.06	0.06	0.05	0.08	0.10	0.05	0.03	0.08
	Training face pose yaw distribution (300-W-LP)								
Random	0.25	0.38	0.37	0.28	0.37	0.35	0.26	0.38	0.38
std dev.	0.01	0.02	0.02	0.02	0.03	0.02	0.03	0.03	0.04
Balanced	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
std dev.	0	0	0	0	0	0	0	0	0
Active	0.16	0.34	0.50	0.15	0.33	0.52	0.20	0.26	0.54
std dev.	0.01	0.03	0.04	0.02	0.02	0.02	0.05	0.02	0.06

Table 3.10 – Comparison of sampling methods for several training set sizes according to the face pose yaw angle. This table reports the NME_{box} of GMDA-S models trained on some 300-W-LP samples and evaluated on the AFLW2000-3D subsets. It also reports the face pose yaw distribution of the training datasets. Active learning favors large face poses, improving the performance of the trained models on this kind of images.

anced training sets for the "Balanced" sampling method. Table 3.10 reports the NME_{box} of the models on the AFLW2000-3D pose subsets. It also reports the face pose distribution of the training datasets sampled from 300-W-LP. The whole 300-W-LP contains 25% low pose images, 37% medium pose images, and 38% large pose images so our "Random" samplings are close to this distribution. "Balanced" samplings have by definition a (33%, 33%, 33%) pose distribution. In the case of "Active" sampling, the table shows that active learning heavily favors the large pose images when selecting samples. Thus, models trained with active learning perform better on AFLW2000-3D large pose subset, especially when training with 48 samples, but a bit worse on the low pose and medium pose subsets. Also, compared to training with the 122,450 images of the whole 300-W-LP, the performance of the models does not degrade much. For example, there is only a 0.6% increase of NME on the large pose subset of AFLW2000-3D when training with 300 samples (0.25% of 300-W-LP size) with active learning compared to training on the whole dataset, meaning the model performs almost the same with 400 times fewer training samples.

3.6 Conclusion

In this Chapter, we have presented the application of our general method to the face alignment task. Our method can successfully re-use the generative power of a pre-trained generative model to predict facial landmark heatmaps using limited (and also sometimes not so accurate) annotated training data. When only 50 training samples are available and the dataset contains very difficult face images such as WFLW, the light-weighted GMDA-R architecture works a bit better but GMDA-S surpasses it otherwise.

The addition of the skip-connections and our proposed hybrid ITL version improve the performance of the GMDA-R architecture, up to 5% (in terms of mean NME decrease) for the skip-connections, and by 3% in average for the hybrid ITL. Also, our proposed acquisition function for active learning, the Negative Neighborhood Magnitude, successfully detects face images where the model struggles to predict accurate landmarks. It makes it possible to select the best samples to annotate and it sometimes halves the number of training samples needed to obtain a similar performance. Our models also work for 3D face alignment when training with limited data. Their performance remain almost the same even when the the number of training samples is divided by 400. However, they might struggle for faces with large pose but using active learning alleviates this issue.

Chapter 4

Application to 3D face reconstruction

As described in Section 1.5, getting good annotations for 3D face reconstruction requires the use of costly equipment such as a scanner which imposes a controlled environment and limits the size and variety of the training dataset. We want to apply our general method to this task to reduce the number of annotated data needed to train a 3D face reconstruction model. In Section 1.5 we also have presented both supervised and self-supervised methods for this task. While self-supervised methods do not require annotations, they tend to predict wrong face scale and head pose.

4.1 Application specificities

4.1.1 Adapting GMDA to 3D face reconstruction

We propose to help self-supervised methods with additional supervised information. We follow our Generative Model Decoder Adaption (GMDA) framework described in Chapter 2 to limit the number of training annotated data needed to add this supervised information. We use the Projected Normalized Coordinate Code [Zhu+16] (PNCC, see Section 1.5.2) as this additional supervised information. The PNCC applies the NCC colormap to the projected vertices of a 3D face. Each color in the NCC colormap corresponds to a position of the 3DMM mean mesh, effectively representing a unique vertex index. By examining the PNCC, it becomes possible to determine the vertex index of each face pixel in the image, enabling a estimation of the head pose and face shape (refer to Figure 4.1). Although the PNCC doesn't encompass all the 3D details of a particular 3D face, such as the exact vertex positions, it still conveys a significant amount of 3D information.

For the 3D face reconstruction application of our method, generating this PNCC becomes our supervised image-to-image translation task. Unlike the application to face

Figure 4.1 – Face images (top) and their corresponding PNCC (bottom).

alignment (Chapter 3) where generating the heatmaps was the final task since the landmark positions can be inferred from them, in this case the PNCC does not contain all the information to reconstruct a 3D face. We propose a two-stage framework: we first use our GMDA method to predict PNCC using only a few annotated data, then we enhance a self-supervised 3D face reconstruction method by adding the predicted PNCC as additional input to the network to help the method predict better head pose and face shape. Figure 4.2 sums up our two-stage framework.

4.1.2 The PNCC predictor

For the PNCC predictor, we use our GMDA-R architecture (see Section 2.2.1) with the original Interleaved Transfer Layers (ITLs) and skip-connections between the encoder and the decoder. We don't use the GMDA-S version because it had trouble generating accurate PNCCs when training with limited data during early experiments. We used the original ITLs because the our proposed two-flow and hybrid versions were not invented yet. Instead of landmark heatmaps like in Chapter 3, this time the last ITL generates the PNCC. The network architecture can be visualized in Figure 4.3.

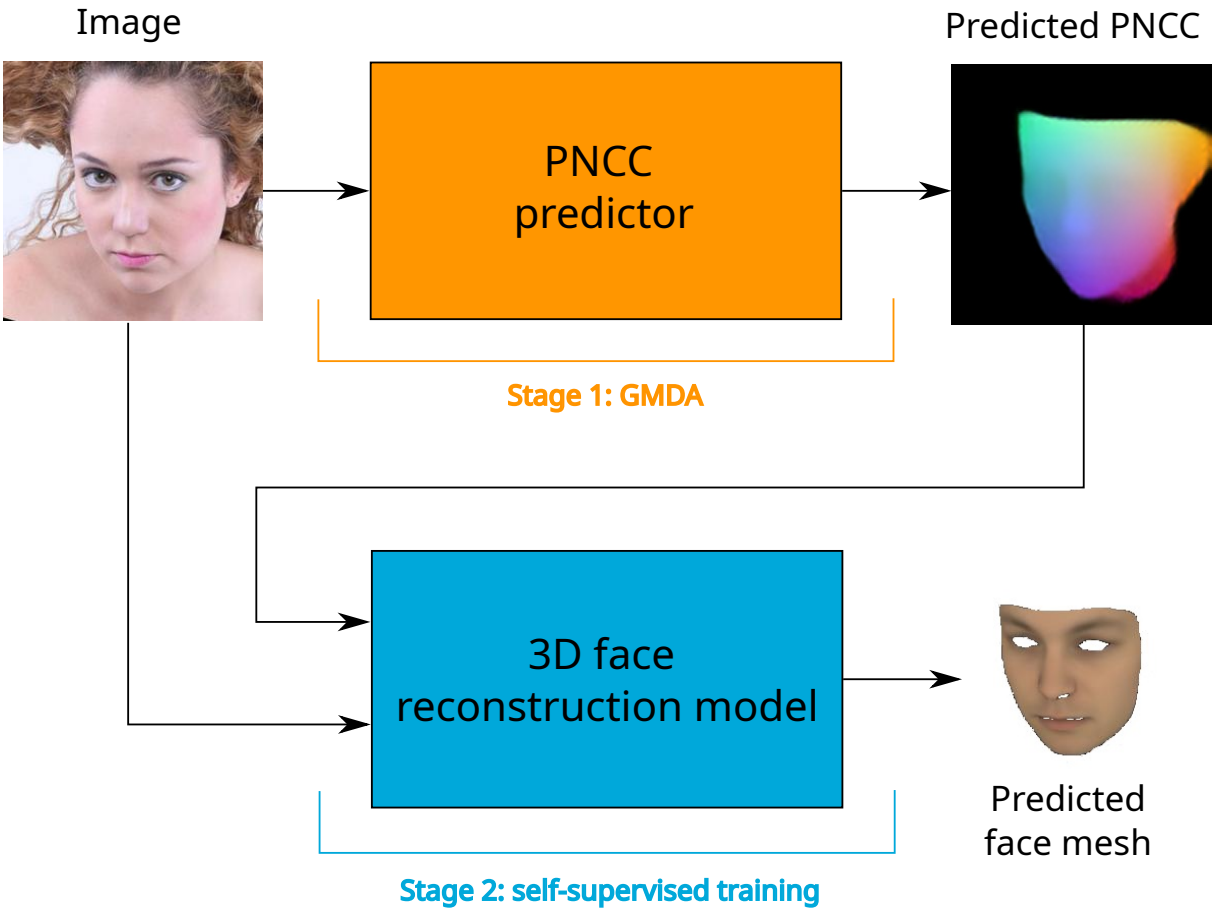


Figure 4.2 – Our two-stage framework for 3D face reconstruction training. We first use our GMDA method to adapt a pre-trained generative model to the PNCC prediction task, using limited annotated data. Then, we train a 3D face reconstruction model with the predicted PNCCs as additional input.

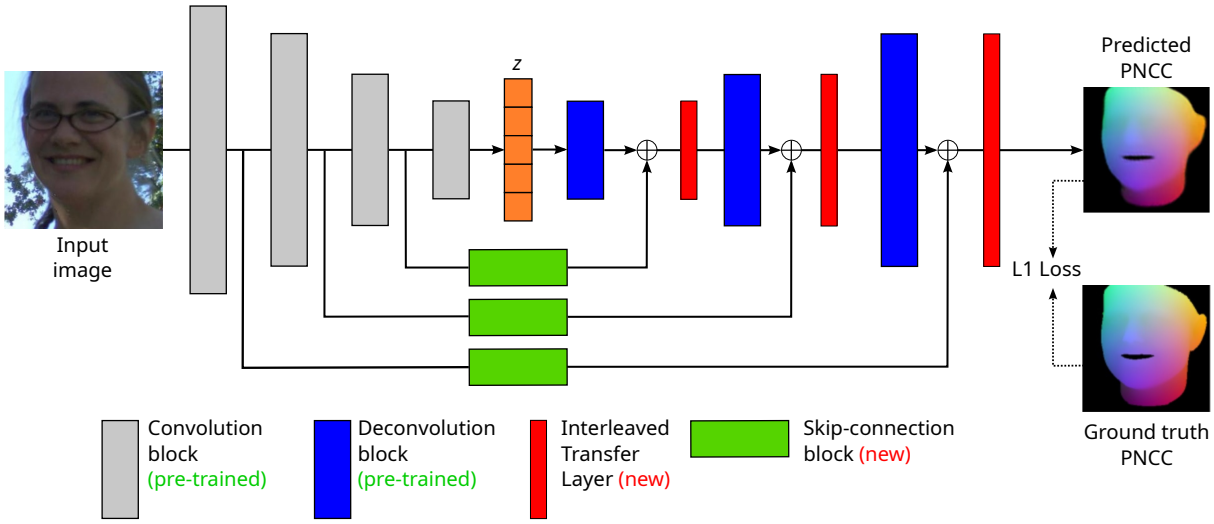


Figure 4.3 – Our PNCC predictor architecture. We use the GMDA-R version with skip-connections. The model is trained to generate the PNCC.

4.1.3 The 3D face reconstruction model

Our self-supervised 3D face reconstruction model is built upon the MoFa framework [Tew+17]. As detailed in Section 1.5.3, the model is composed of a neural network that predicts a vector $\mathbf{p}$ containing all necessary parameters for reconstructing a facial image, including 3DMM parameters, illumination, and head pose. To render the face, a differentiable renderer is utilized, allowing for the back-propagation of training loss. We keep the same architecture and training loss as MoFa, except for one modification: the predicted PNCC (from our PNCC predictor) is stacked channel-wise with the face image at the network input (visualized in Figure 4.4), improving the network ability to accurately predict head pose and face shape.

4.2 Experimental settings

4.2.1 Training datasets

We train our PNCC predictor on the 300-W-LP dataset [Zhu+16]. This dataset contains 3D landmarks annotation (see Chapter 3) but also 3D face reconstruction annotations obtained through 3DMM fitting [RV05]. Since the 3DMM annotations are obtained using optimization, their quality is variable but the dataset comes with the benefit of

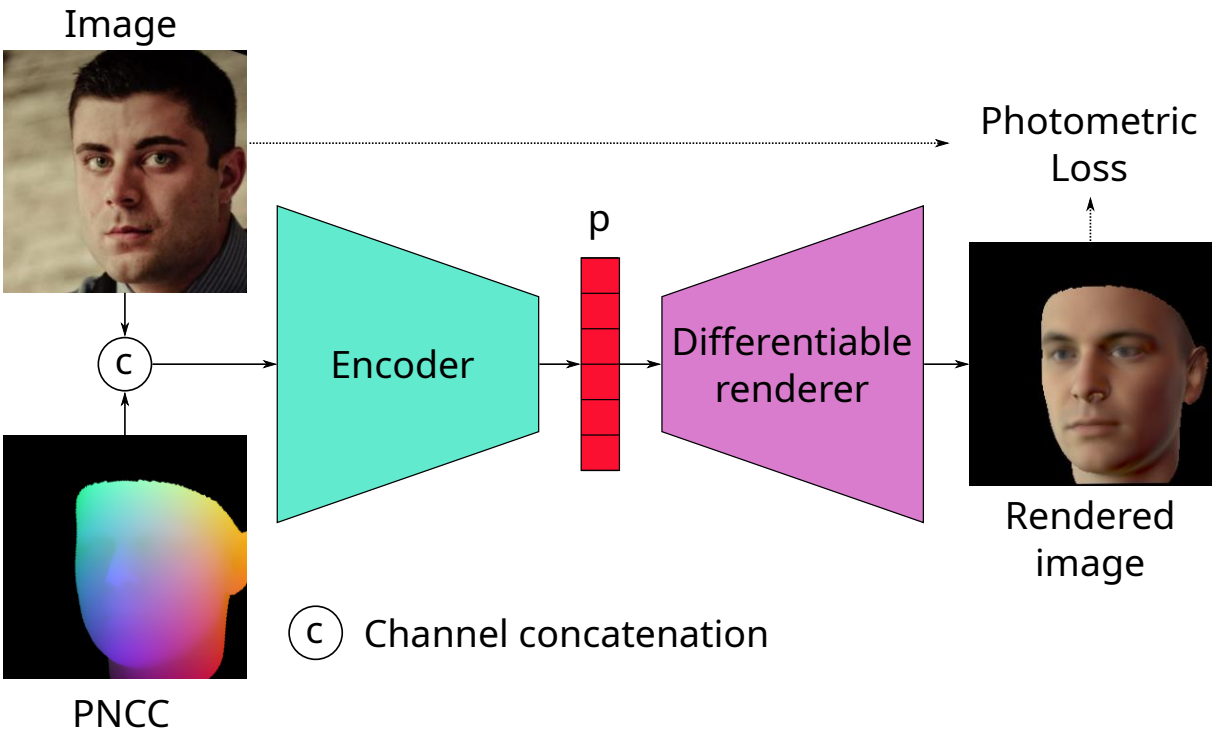


Figure 4.4 – Our 3D face reconstruction architecture. As input, we stack (channel-wise) the face image with the PNCC predicted from our PNCC predictor. The encoder predicts the parameter vector $\mathbf{p}$. The differentiable renderer renders from $\mathbf{p}$ the reconstructed face image.

having a large number of samples (122,450 face images). The 3D face annotations use the Basel face topology [Pay+09].

We train our 3D face reconstruction model on CelebA [Liu+15]. This dataset contains more than 200,000 face images of celebrities. Once our PNCC predictor is trained, we use it to annotate this dataset with PNCC annotations.

4.2.2 Architectures and training parameters

4.2.2.1 PNCC predictor

As stated in Section 4.1.2, we use the GMDA-R architecture equipped with the Interleaved Transfer Layers (original version) with skip-connections between the encoder and the decoder for our PNCC predictor. The architecture is similar to the GMDA-R one used for face alignment (see Chapter 3). The encoder is a ResNet-18 and the decoder is an inverted ResNet-18 with convolutions replaced with deconvolutions, both pre-trained using the 3FabRec [BW20] weights available at `https://github.com/browatbn2/3FabRec`. The input and output image size is 256×256. The ITLs are 3×3 convolution layers and for the skip-connections blocks, we use the hierarchical, parallel, and multi-scale block from [BT17a]. We use 5 ITLs and output PNCC of size 128×128. As training loss we use the L1 error between the predicted and ground truth PNCCs. The batch size is 8. We train from scratch the ITLs and the skip-connections blocks using the Adam optimizer [KB15] with a fixed learning rate of 4e-4. In the same time, we fine-tune the encoder using again the Adam optimizer with a fixed learning rate of 2e-5. The decoder layers weights are frozen. As data augmentations, we use random rotations, translations, contrast changes, brightness changes and occlusions. The number of training steps depends on the training set size and is chosen using cross-validation.

4.2.2.2 3D face reconstruction model

For the 3D face reconstruction, the encoder is a ResNet-50 [He+16] pre-trained on ImageNet [Den+09]. Before being processed by the encoder, the input face images are first cropped and aligned using a landmark detector [Guo+20]. We use the same training losses as in MoFa [Tew+17], we train the network for 200 epochs using the Adam optimizer with a learning rate of 4e-5 and a batch size of 128.

4.2.3 Evaluation metrics

We are interested into evaluating the predicted face shape but also the predicted head pose. To do so, we use several metrics. To evaluate the face shape we use the Normalized Mean Error (NME):

$$\text{NME} = \frac{1}{N} \sum_{i=1}^{N} \frac{||\mathbf{v}_i - \hat{\mathbf{v}}_i||_2}{d} \ , \tag{4.1}$$

where $\mathbf{v}_i$ and $\hat{\mathbf{v}}_i$ are the ground truth and predicted vertices positions respectively. N is the number of vertices and d is a normalization distance. From the NME we can derive several metrics. The *2D Dense Alignment* metric computes the NME on the 2D positions of the vertices. The *3D Dense Alignment* metric is the NME on the 3D positions of the vertices. Following common practices [Fen+18a; Guo+20; Rua+21], for both metrics the normalization distance is the face bounding box size. We also compute a third metric called *3D Face Reconstruction* metric. This time, the predicted face mesh is first aligned with the ground truth mesh using Procrustes Analysis before computing the NME. In this case, again following common practices [Fen+18a; Guo+20; Rua+21], we use the 3D interocular distance (3D distance between the outer eye corners) as normalization distance. Unlike the previous two, this metric is invariant to the predicted head pose, it only evaluates the predicted face shape.

To better evaluate the predicted head orientation, we also compute the Mean Absolute Error (MAE) (see Equation 4.2) between the predicted $\hat{\mathbf{p}}$ and ground truth $\mathbf{p}$ rotation parameters. Since our model operates on aligned images, we only compute the MAE for the yaw angle (invariant to the alignment).

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{p}_i - \hat{\mathbf{p}}_i| \tag{4.2}$$

4.2.4 Evaluation dataset

We assess the 3D face reconstruction task using the AFLW2000-3D dataset [Zhu+16] as evaluation dataset. This dataset was created by re-annotating the initial 2,000 images of AFLW [Koe+11] with annotations consistent with 300-W-LP. While many recent self-supervised methods evaluate their performance on the NoW dataset [San+19], this dataset does not evaluate the predicted head pose because predicted face meshes are aligned with ground truth prior to computing metrics. In contrast, our objective is to showcase the

enhanced performance of predicted head pose with the addition of the PNCC input. Hence, we choose to employ the AFLW2000-3D dataset for our evaluation.

4.3 Results

4.3.1 PNCC prediction

In Figure 4.5, a comparison is presented between the ground truth PNCC of selected images from AFWL2000-3D [Zhu+16] and the predictions generated by our PNCC predictor. The figure shows the results of two models: one trained on the entire 300-W-LP dataset [Zhu+16] referred to as PNCC_{full}, and another trained on only 50 samples from 300-W-LP, denoted as PNCC_{few}. Due to the semi-automatic annotation process used for AFLW2000-3D and the limited shape and expression spaces of the 3DMM, some of the PNCC annotations may not be entirely accurate. Thus, our model predictions are sometimes better than ground truth annotations. When the model is trained on a substantially smaller subset of the 300-W-LP dataset, specifically just 50 samples (0.04% of the total dataset), the predictions become slightly blurrier compared to the PNCC_{full} model. However, despite this heavy reduction of the training data, the head pose and overall facial shape predictions remain generally accurate in most cases.

4.3.2 3D face reconstruction

We compare our self-supervised model with MoFa [Tew+17], adopting the same architecture and training parameters, with the exception of the PNCC input. A direct comparison with newer self-supervised models is challenging due to the prevalent use of the FLAME face topology [Li+17] in most recent models [San+19; Fen+21], whereas our model, based on MoFa, and the AFLW2000-3D annotations use the Basel face topology [Pay+09]. The primary objective of this section is to show the performance improvement achieved by incorporating PNCC as an additional input into a 3D face reconstruction model. While this principle could theoretically extend to more modern methods, the lack of an evaluation dataset under the FLAME topology, which also includes head pose prediction evaluation, prevented us from extending the comparison.

Table 4.1 shows the results of dense alignment, 3D face reconstruction and head pose estimation evaluation conducted on the AFLW2000-3D dataset [Zhu+16]. The nomenclature MoFaPNCC_{few} represents our 3D face reconstruction model trained with predictions

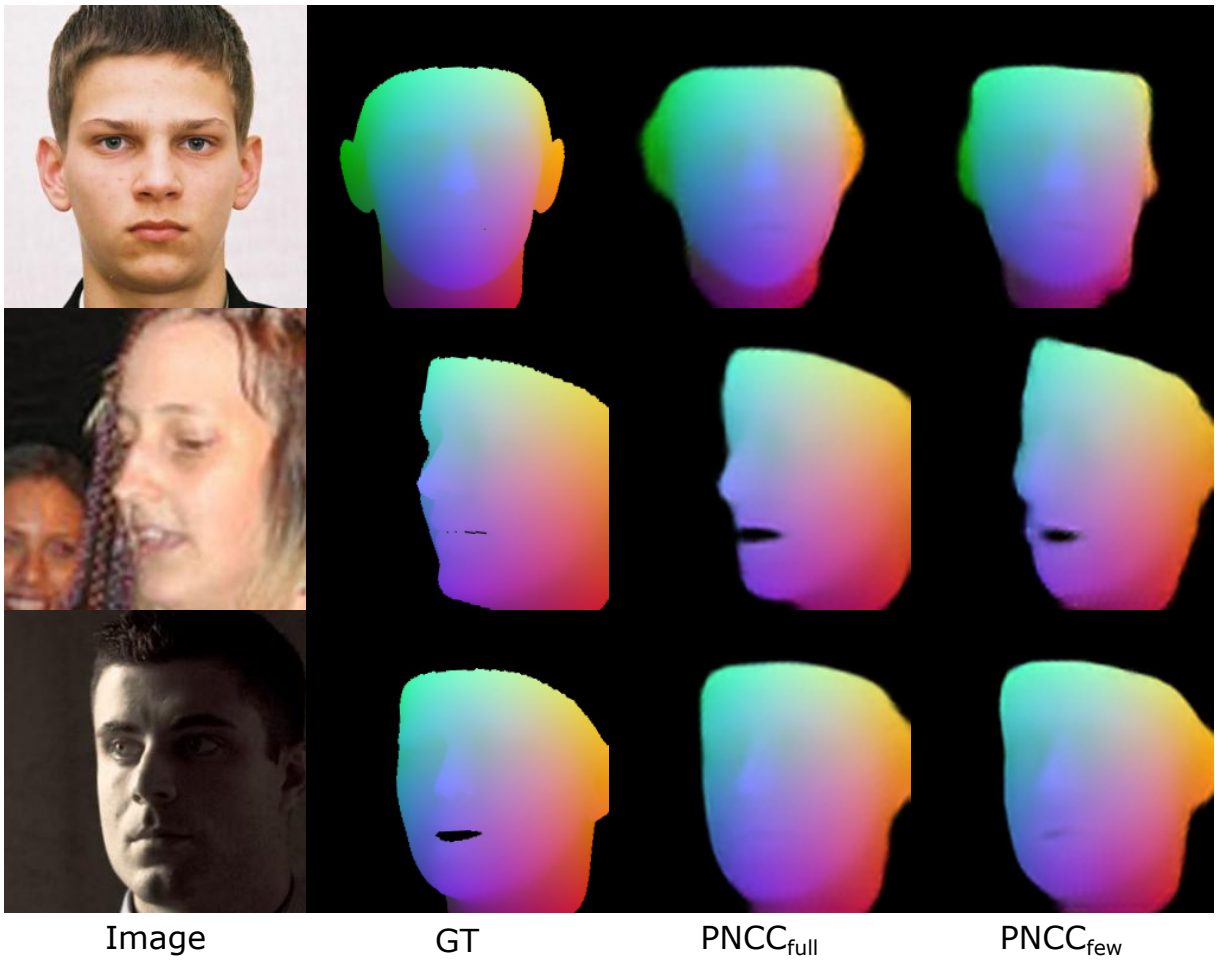


Figure 4.5 – Comparison between ground truth and predicted PNCCs of some images of AFLW2000-3D. Far left: input images. Middle left: ground truth PNCCs. Middle right: PNCCs predicted by our model PNCC$_{full}$ trained on the whole 300-W-LP dataset. Far right: PNCCs predicted by our model PNCC$_{few}$ trained on only 50 samples of 300-W-LP (0.04% of the dataset). Our predictions are sometimes even better than the ground truth (the mouth openings of the middle and bottom images are wrong in the ground truth).

from PNCC_{few}, while MoFaPNCC_{full} means that the model has been trained using PNCC predictions from PNCC_{full}. In the evaluation phase on AFLW2000-3D, the second and third rows utilize the predictions of PNCC_{few} and PNCC_{full} respectfully. Regarding the dense alignment metrics, both MoFaPNCC_{full} and MoFaPNCC_{few} yield superior results in comparison to the original MoFa [Tew+17]. For MoFaPNCC_{full} the 2D and 3D dense alignment errors are reduced by 4% and 6% respectively. For MoFaPNCC_{few}, the reductions is around 3% for both metrics.

These results prove that augmenting the model input with PNCC leads to enhanced head pose predictions, even in instances where PNCC predictions are not flawless. In terms of the 3D face reconstruction metric, which includes a rigid alignment prior to error computation, the baseline MoFa exhibits slightly better performance. It's important to note that this metric doesn't consider the predicted head pose, whereas our architecture's core objective revolves around improving the predicted head pose through the inclusion of PNCC in the input.

We further explore the impact of PNCC quality at test time. The results of these investigations are presented in the lower segment of Table 4.1. Our evaluation still uses the two 3D face reconstruction models, denoted as MoFaPNCC_{few} and MoFaPNCC_{full}, but with variations in the PNCC input used during testing. Specifically, we assess the models with predictions from PNCC_{few}, PNCC_{full}, and ground truth PNCCs (PNCC_{GT}).

For the dense alignment metrics, interesting patterns emerge. MoFaPNCC_{few} yields nearly equivalent outcomes to MoFaPNCC_{full} when both models are evaluated using the predictions from PNCC_{full}. This finding underscores that even if a model is trained with suboptimal PNCCs, its performance benefits from access to superior PNCCs during testing. Conversely, when both models are tested using the predictions from PNCC_{few}, MoFaPNCC_{full} demonstrates notably poorer results compared to MoFaPNCC_{few}. Employing ground truth PNCCs considerably enhances dense alignment results for both models.

The results for the 3D face reconstruction metric are more puzzling. Unlike the dense alignment metrics, utilizing PNCC_{full} alongside MoFaPNCC_{few} leads to inferior outcomes, while coupling PNCC_{few} with MoFaPNCC_{full} improves the metric. Intriguingly, employing ground truth PNCCs enhances results for MoFaPNCC_{full} but doesn't exhibit the same effect for MoFaPNCC_{few}. All these observations suggest that the PNCC predominantly conveys information related to head pose rather than accurate facial geometry which is evaluated by the 3D face reconstruction metric.

Method	Dense 2D	Dense 3D	Face. Rec.	Yaw MAE
MoFa [Tew+17]	4.31	5.85	7.49	4.97
MoFaPNCC$_5$ w/ PNCC$_{few}$	4.20	5.66	7.61	4.95
MoFaPNCC$_{full}$ w/ PNCC$_{full}$	4.12	5.48	7.55	4.66
MoFaPNCC$_{few}$ w/ PNCC$_{full}$	4.14	5.50	7.93	4.76
MoFaPNCC$_{full}$ w/ PNCC$_{few}$	4.69	6.59	7.05	6.14
MoFaPNCC$_{few}$ w/ PNCC$_{GT}$	3.82	5.08	7.73	4.26
MoFaPNCC$_{full}$ w/ PNCC$_{GT}$	3.58	4.88	7.46	3.93

Table 4.1 – Dense alignment, 3D face reconstruction and head pose metrics on AFLW2000-3D. MoFaPNCC$_{full}$ has been trained with PNCC$_{full}$ predictions and MoFaPNCC$_{few}$ with PNCC$_{few}$. Bottom part displays results of our models depending on the PNCCs used at test time. PNCC$_{GT}$ denotes the ground truth PNNCs of AFLW2000-3D.

4.3.3 Head pose rotation estimation

The results concerning head pose rotation estimation are detailed in the last column of Table 4.1. As previously mentioned in Section 4.2.2, our model functions on aligned images, thereby we only present the Mean Absolute Error (MAE) for the yaw angle which is unaffected by the alignment process. We observe outcomes in line with those of the dense alignment metrics. The incorporation of PNCC yields enhancements in the predicted yaw angle. The Yaw MAE metric is reduced by 6% with the MoFaPNCC$_{full}$ compared to the original MoFa. The improvement is relatively modest for MoFaPNCC$_{few}$ when evaluated with the predictions from PNCC$_{few}$ (0.4% MAE reduction) but only 50 annotated training samples have been used in this configuration. Similar to the dense alignment metrics, utilizing superior PNCCs during testing leads to enhancements in results for both MoFaPNCC$_{few}$ and MoFaPNCC$_{full}$.

4.3.4 Qualitative results

Figure 4.6 visually displays the face meshes, focusing solely on geometry without texture, generated by MoFa [Tew+17] and our models across various facial images. Notably, our models exhibit improved predictions for head pose parameters, specifically the face yaw angle in the top image and face scale in the middle and bottom images.

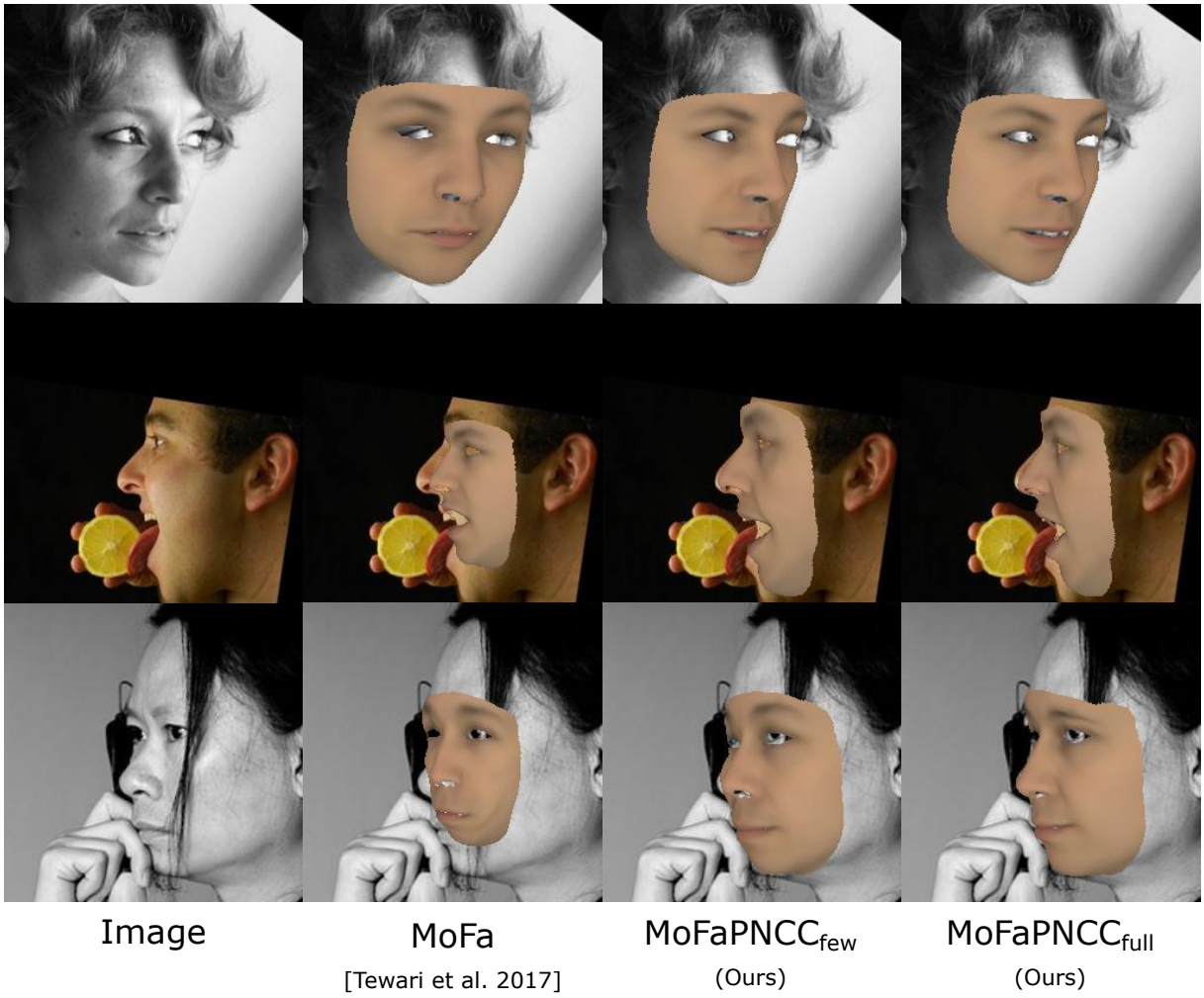


Figure 4.6 – Comparison of some 3D face reconstruction predictions (only geometry). Our models predict better face scale (middle and bottom images) and rotation parameters (top and bottom images).

4.4 Conclusion

In this Chapter, we have applied our general method to improve 3D face reconstruction. We used our GMDA framework to adapt a pre-trained generative model to the PNNC prediction task. Even with limited, and not so accurate, training data, our model can predict decent PNCCs, sometimes better than the ground truth. This proves that our method is not restricted to facial landmark heatmaps (see Chapter 3) and can be applied to other facial image-to-image translation tasks. Our experiments shows that adding the PNCC to the input of a self-supervised 3D face reconstructions improves the predicted head pose, even with PNNCs predicted by a PNNC predictor trained with only 50 annotated samples.

Conclusion and perspectives

Conclusion

The goal of this PhD was to propose an original approach to overcome the issue of the lack of annotated training data which plagues many possible applications of Deep Learning. Among these applications, we focused on facial analysis tasks, which is a domain of interest for InterDigital, because many of them suffer from this problem by either lack of annotated data or having only access to poor annotations. This degrades the quality of the models trained on this kind of data.

To resolve this issue, we based our work on transfer learning from self-supervised models. This kind of models learns, using pretext tasks, from non-annotated data which can be found in large quantities. Hence they learn powerful representations robust to many factors of variations. These learned representations can then be used for other applications with only a few annotated data needed to adapt them to the downstream task. While this approach is not novel, most existing models use for transfer learning only low-dimensional representations derived from self-supervised encoder-like models. Even in the case of generative models, usually only the encoder is kept during the transfer learning and the decoder is discarded.

In this PhD, we proposed a new approach: the Generative Model Decoder Adaptation (GMDA) which performs transfer learning using high-dimensional features from a self-supervised generative model decoder. This approach is particularly well suited for downstream tasks where the model prediction must be a high-dimensional value such as supervised image-to-image translation tasks, since it avoids the need to train a large neural network for the downstream task which would require many annotated training samples. Unlike methods which only use the low-dimensional representations, our approach necessitates to make creative architecture changes inside the generative model, and particularly the decoder, during the transfer learning to adapt it to the supervised image-to-image translation task.

Evaluation of the proposed architecture options

We proposed several architecture options for this approach. For the generative model, we tried a ResNet [He+16] autoencoder (GMDA-R version) and StyleGAN [KLA19] (GMDA-S version). To adapt the generative model, we took inspiration from Interleaved Transfer Layers [BW20] which are convolutional layers interleaved with the decoder layers, but we also proposed two improved versions of them: the two-flow ITL and the hybrid ITL. Adding skip-connections between the encoder and the decoder was also tested.

We tested these architecture choices on our first facial analysis task: face alignment. This task aims to predict the positions of facial anatomical landmarks on a face. This task can be seen as an image-to-image translation task if the positions of the landmarks are encoded into heatmaps so we applied our GMDA method to this task.

Each architecture option was tested with different training set sizes, sometimes also on different datasets, to see how this particular option behaves depending on the number of available training samples. Each reported result is the mean of 5 runs with random initial parameters and random training samples to account for the possible statistical variance of the tested setting. Thus, all the presented results necessitated hundreds of runs. From our experiments with our different architectures we could draw several conclusions.

1. For almost all tested datasets and training set sizes, the GMDA-S architecture with its increased capacity and generative quality works better than the light-weighted GMDA-R models except when only a few annotated samples are available for training (GMDA-S is always better in our experiments when the training size is strictly superior to 50) and the test face images are very challenging.
2. Skip-connections between the encoder and the decoder improve the accuracy of the model for the GMDA-R architecture, up to 5% in terms of NME reduction, but do not seem to work well for the GMDA-S architecture. This might be due to StyleGAN unusual way to feed the latent code to the generator.
3. Our proposed hybrid ITL improves the performance of the GMDA-R models with skip-connections compared to the original ITL with a 3% NME reduction in average.

Face alignment results

We also proposed a novel acquisition function, the *Negative Neighborhood Magnitude*, for active learning which assesses the quality of the predicted heatmaps. Thanks to this

function, when constructing a face alignment dataset, one can select the best samples to annotate rather than annotating random face images, which improves the performance of the model with equal number of training samples. In some cases, a model trained with active learning obtained better performance compared to a model trained with the double of its number of training samples but without active learning.

Annotating an image for face alignment is time-consuming so face alignment datasets are usually quite small which makes the learned model prone to overfitting. Also, annotations can be ambiguous (e.g. the positions of the landmarks on the outline of the face) which may lead to inconsistent annotations again hurting the trained model performance. However, using our method with the proposed architecture and training scheme improvements, we were able to successfully adapt the generative model to face alignment using only limited annotated data during training, even only 50 samples, outperforming state-of-the-art for many low data settings on many datasets. For example, on the AFLW dataset, our models outperform all other existing methods if the training set size is inferior to 10% of the whole training dataset size. In the case of 3D face alignment, for some settings, even if we divide the number of training samples by 400, the performance of our model remains almost the same.

3D face reconstruction results

The other task on which we experimented our approach was 3D face reconstruction which aims to retrieve the 3D face rig parameters from a single monocular face image. This task also lacks large and accurately annotated dataset because 3D face annotations require the use of face scanner or multi-camera setup which imposes a controlled environment. Most existing training datasets are annotated in a semi-automatic way which leads to poor annotations.

Rather than directly predicting the 3D face rig parameters, we used our method to improve self-supervised methods which tends to predict wrong head pose of face scale due to the lack of 3D annotations during their training. We proposed to add supervised information, through the use of the PNCC [Zhu+16], to the input of the self-supervised model, in addition to the face image.

We used our GMDA framework to train a PNCC predictor using limited annotated data. Once our predictor trained, we annotated a face dataset with PNCCs and trained the self-supervised 3D face reconstruction on the augmented dataset. As PNCC predictor, we used our GMDA-R architecture with Interleaved Transfer Layers (original version) and

skip-connections between the encoder and the decoder. Even when training with only 50 samples, our predictor could predict decent PNCCs, sometimes even better than the ground truth annotations.

Our experiments proved that the self-supervised 3D face reconstruction model indeed benefits from the PNNC information and predicts better head pose and face scale. Both the 3D dense alignment metric and the Yaw MAE are reduced by up to 6% compared to the original self-supervised method. However, the PNCC does not seem to improve the predicted face geometry which indicates that the PNCC mostly contains head pose information. Our experiments also showed that even models trained on not so accurate PNCCs benefit from having access to better PNCCs at test time.

Perspectives

The self-supervised 3D face reconstruction method [Tew+17] used as baseline in Chapter 4 is relatively old but we chose it because recent methods use the FLAME face topology [Li+17] which is different from the one, based on Basel [Pay+09], used in our selected annotated dataset AFLW2000-3D. If we had time, we would have liked to convert the dataset annotations to the FLAME topology to see if the predicted head pose improvement provided with the addition to the PNCC still holds for recent self-supervised 3D face reconstruction methods.

We applied our method to two facial analysis tasks: face alignment and 3D face reconstruction but it could be interesting to test it for other supervised image-image translation tasks such as semantic face segmentation. Also, expanding the approach beyond the facial domain with images of other kinds of objects such as animals, buildings or even scene images containing multiple objects could also work. Could it also be applied to images with symbols (text documents, music sheets, ...)? Generative models already exist for these kinds of images so it should be possible to adapt them to the downstream task if they share similar structures as the ones used in this PhD.

Also, even though our methodology is initially only applicable to image-to-image translation tasks, we have seen in Chapter 4 that by using intermediary image-like representation such as the PNCC, our method can also be used to help during the training of tasks which are not image-to-image translation tasks like 3D face reconstruction. Thus, our method might be applicable to tasks such as object detection or image classification if we can find an image-like representation of the target data.

For the generative model, we only experimented on convolutional networks but many recent models are based on transformers [Zha+21; Zha+22] which have shown great performance in many domains. Whether and how they can be adapted to another image-to-image translation task, like we did for GMDA-R and GMDA-S, is an open question. Especially, is it possible to interleave new layers between the decoder blocks like we did with Interleaved Transfer Layers? Or other strategies must be applied to re-use the decoder layers? Since convolutional layers are not used in transformers, simply interleaving the decoder blocks with linear layers could be a first experiment to test.

Finally, self-supervised models can learn powerful representations using large non annotated datasets. The success of models such as ChatGPT, and most of our experiments in this PhD are proofs of that. But there is still no guaranty that these representations are useful for any downstream task. In the domain of computer vision, with the progress of computer graphics, we think that the increasing quality of synthetic samples may be another solution to handle the lack of annotated samples. While synthetic data is not without limitations such as potential domain shift and limited variety, improving on these two factors might be the key to obtain large annotated datasets almost as good a real ones and make supervised training relevant for tasks that currently lack annotated data.

Bibliography

Author's publications

International conferences with peer-review

Articles

[Dor+22] Martin Dornier, Philippe-Henri Gosselin, Christian Raymond, Yann Ricquebourg, and Bertrand Coüasnon, « SCAF: Skip-Connections in Auto-encoder for Face Alignment with Few Annotated Data », *in*: *International Conference on Image Analysis and Processing*, Springer, 2022, pp. 425–437.

Posters

[Dor+23] Martin Dornier, Philippe-Henri Gosselin, Christian Raymond, Yann Ricquebourg, and Bertrand Coüasnon, « Improving self-supervised 3D face reconstruction with few-shot transfer learning », *in*: *MIG 2023: 16th annual ACM/SIGGRAPH conference on Motion, Interaction and Games*, 2023.

Journal article being finalized, to be submitted to Pattern Recognition Letters

[Dor+] Martin Dornier, Philippe-Henri Gosselin, Christian Raymond, Yann Ricquebourg, and Bertrand Coüasnon, « StyleGAN-based heatmap generator for face alignment with limited training data », *in*: ().

References

[APC21] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or, « Restyle: A residual-based stylegan encoder via iterative refinement », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6711–6720.

[BAP14] Philip Bachman, Ouais Alsharif, and Doina Precup, « Learning with pseudo-ensembles », *in*: *Advances in neural information processing systems* 27 (2014).

[BST21] Adrian Bulat, Enrique Sanchez, and Georgios Tzimiropoulos, « Subpixel Heatmap Regression for Facial Landmark Localization », *in*: *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

[BT17a] Adrian Bulat and Georgios Tzimiropoulos, « Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3706–3714.

[BT17b] Adrian Bulat and Georgios Tzimiropoulos, « How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks) », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1021–1030.

[Bul+22] Adrian Bulat, Shiyang Cheng, Jing Yang, Andrew Garbett, Enrique Sanchez, and Georgios Tzimiropoulos, « Pre-training strategies and datasets for facial representation learning », *in*: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, Springer, 2022, pp. 107–125.

[BV03] Volker Blanz and Thomas Vetter, « Face recognition based on fitting a 3D morphable model », *in*: *IEEE Transactions on pattern analysis and machine intelligence* 25.*9* (2003), pp. 1063–1074.

[BW20] Bjorn Browatzki and Christian Wallraven, « 3fabrec: Fast few-shot face alignment by reconstruction », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6110–6120.

[Cao+14] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun, « Face alignment by explicit shape regression », *in*: *International journal of computer vision* 107.*2* (2014), pp. 177–190.

[Car+20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, « Unsupervised learning of visual features by contrasting cluster assignments », *in*: *Advances in neural information processing systems* 33 (2020), pp. 9912–9924.

[Cha+21] Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang, « Ensembling with deep generative views », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14997–15007.

[Che+20a] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, « A simple framework for contrastive learning of visual representations », *in*: *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[Che+20b] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton, « Big self-supervised models are strong semi-supervised learners », *in*: *Advances in neural information processing systems* 33 (2020), pp. 22243–22255.

[Chi+22] Darius Chira, Ilian Haralampiev, Ole Winther, Andrea Dittadi, and Valentin Liévin, « Image super-resolution with deep variational autoencoders », *in*: *European Conference on Computer Vision*, Springer, 2022, pp. 395–411.

[Coo+95] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham, « Active shape models-their training and application », *in*: *Computer vision and image understanding* 61.*1* (1995), pp. 38–59.

[CV95] Corinna Cortes and Vladimir Vapnik, « Support-vector networks », *in*: *Machine learning* 20.*3* (1995), pp. 273–297.

[DBC19] Arnaud Dapogny, Kevin Bailly, and Matthieu Cord, « Decafa: Deep convolutional cascade for face alignment in the wild », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6893–6901.

[Den+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, « Imagenet: A large-scale hierarchical image database », *in*: *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[Den+19] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong, « Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set », *in*: *IEEE Computer Vision and Pattern Recognition Workshops*, 2019.

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding », *in*: *arXiv preprint arXiv:1810.04805* (2018).

[DGE15] Carl Doersch, Abhinav Gupta, and Alexei A. Efros, « Unsupervised Visual Representation Learning by Context Prediction », *in*: *International Conference on Computer Vision (ICCV)*, 2015.

[DKD17] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell, « Adversarial Feature Learning », *in*: *International Conference on Learning Representations*, 2017.

[Don+15] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, « Image super-resolution using deep convolutional networks », *in*: *IEEE transactions on pattern analysis and machine intelligence* 38.*2* (2015), pp. 295–307.

[Don+18] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang, « Style aggregated network for facial landmark detection », *in*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 379–388.

[DS19] Jeff Donahue and Karen Simonyan, « Large scale adversarial representation learning », *in*: *Advances in Neural Information Processing Systems*, 2019, pp. 10542–10552.

[DT05] Navneet Dalal and Bill Triggs, « Histograms of oriented gradients for human detection », *in*: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, Ieee, 2005, pp. 886–893.

[Dum+17] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville, « Adversarially Learned Inference », *in*: *International Conference on Learning Representations*, 2017.

[DY19] Xuanyi Dong and Yi Yang, « Teacher supervises students how to learn from partially labeled images for facial landmark detection », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 783–792.

[Fen+18a] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou, « Joint 3d face reconstruction and dense alignment with position map regression network », *in*: *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 534–551.

[Fen+18b] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu, « Wing loss for robust facial landmark localisation with convolutional neural networks », *in*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2235–2245.

[Fen+21] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart, « Learning an animatable detailed 3D face model from in-the-wild images », *in*: *ACM Transactions on Graphics (ToG)* 40.*4* (2021), pp. 1–13.

[GIG17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani, « Deep bayesian active learning with image data », *in*: *International Conference on Machine Learning*, PMLR, 2017, pp. 1183–1192.

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, « Generative Adversarial Nets », *in*: *Advances in Neural Information Processing Systems 27*, ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Curran Associates, Inc., 2014, pp. 2672–2680.

[GSZ20] Jinjin Gu, Yujun Shen, and Bolei Zhou, « Image processing using multi-code gan prior », *in*: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3012–3021.

[Guo+20] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li, « Towards Fast, Accurate and Stable 3D Dense Face Alignment », *in*: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[HB17] Xun Huang and Serge Belongie, « Arbitrary style transfer in real-time with adaptive instance normalization », *in*: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, « Deep residual learning for image recognition », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[He+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick, « Masked autoencoders are scalable vision learners », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16000–16009.

[Hon+18] Sina Honari, Pavlo Molchanov, Stephen Tyree, Pascal Vincent, Christopher Pal, and Jan Kautz, « Improving landmark localization with semi-supervised learning », *in*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1546–1555.

[Hua+21] Yangyu Huang, Hao Yang, Chong Li, Jongyoo Kim, and Fangyun Wei, « Adnet: Leveraging error-bias towards normal direction in face alignment », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3080–3090.

[JOL21] Harim Jung, Myeong-Seok Oh, and Seong-Whan Lee, « Learning free-form deformation for 3D face reconstruction from in-the-wild images », *in*: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2021, pp. 2737–2742.

[Kar+20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, « Analyzing and improving the image quality of stylegan », *in*: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[Kar+21] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, « Alias-free generative adversarial networks », *in*: *Advances in Neural Information Processing Systems* 34 (2021).

[KB15] Diederik P. Kingma and Jimmy Ba, « Adam: A Method for Stochastic Optimization », *in*: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun, 2015, URL: `http://arxiv.org/abs/1412.6980`.

[KKC21] Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho, « GAN Inversion for Out-of-Range Images with Geometric Transformations », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13941–13949.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila, « A style-based generator architecture for generative adversarial networks », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[Koe+11] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof, « Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization », *in*: *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, IEEE, 2011, pp. 2144–2151.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, « Imagenet classification with deep convolutional neural networks », *in*: *Advances in neural information processing systems* 25 (2012).

[Kum+20] Abhinav Kumar, Tim K Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng, « LUVLi Face Alignment: Estimating Landmarks' Location, Uncertainty, and Visibility Likelihood », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8236–8246.

[KVG19] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal, « Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning », *in*: *Advances in neural information processing systems* 32 (2019), pp. 7026–7037.

[KW14] Diederik P. Kingma and Max Welling, « Auto-Encoding Variational Bayes », *in*: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun, 2014, URL: `http://arxiv.org/abs/1312.6114`.

[LeC98] Yann LeCun, « The MNIST database of handwritten digits », *in*: *http://yann.lecun. com/exdb/mnist/* (1998).

[Led+17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., « Photo-realistic single image super-resolution using a generative adversarial network », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[Lee+13] Dong-Hyun Lee et al., « Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks », *in*: *Workshop on challenges in representation learning, ICML*, vol. 3, 2, Atlanta, 2013, p. 896.

[Li+17] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero, « Learning a model of facial shape and expression from 4D scans. », *in*: *ACM Trans. Graph.* 36.*6* (2017), pp. 194–1.

[Li+22] Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang, « Clip-event: Connecting text and images with event structures », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16420–16429.

[Li+23] Chunlu Li, Andreas Morel-Forster, Thomas Vetter, Bernhard Egger, and Adam Kortylewski, « Robust Model-based Face Reconstruction through Weakly-Supervised Outlier Segmentation », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 372–381.

[Liu+15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, « Deep learning face attributes in the wild », *in*: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[Llo82] S. Lloyd, « Least squares quantization in PCM », *in*: *IEEE Transactions on Information Theory* 28.*2* (1982), pp. 129–137, DOI: 10.1109/TIT.1982.1056489.

[Low04] David G Lowe, « Distinctive image features from scale-invariant keypoints », *in*: *International journal of computer vision* 60 (2004), pp. 91–110.

[Mak+16] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow, « Adversarial Autoencoders », *in*: *International Conference on Learning Representations*, 2016, URL: http://arxiv.org/abs/1511.05644.

[MB04] Iain Matthews and Simon Baker, « Active appearance models revisited », *in*: *International journal of computer vision* 60.*2* (2004), pp. 135–164.

[Mül66] Claus Müller, *Spherical harmonics*, Springer, 1966.

[NF16] Mehdi Noroozi and Paolo Favaro, « Unsupervised learning of visual representations by solving jigsaw puzzles », *in*: *European conference on computer vision*, Springer, 2016, pp. 69–84.

[Nit+22] Yotam Nitzan, Rinon Gal, Ofir Brenner, and Daniel Cohen-Or, « LARGE: Latent-Based Regression through GAN Semantics », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19239–19249.

[NYD16] Alejandro Newell, Kaiyu Yang, and Jia Deng, « Stacked hourglass networks for human pose estimation », *in*: *European conference on computer vision*, Springer, 2016, pp. 483–499.

[Pay+09] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter, « A 3D Face Model for Pose and Illumination Invariant Face Recognition », *in*: *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009, pp. 296–301, DOI: 10.1109/AVSS.2009.58.

[Qia+19] Shengju Qian, Keqiang Sun, Wayne Wu, Chen Qian, and Jiaya Jia, « Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10153–10163.

[Rad+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., « Improving language understanding by generative pre-training », *in*: (2018).

[Ric+21] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or, « Encoding in style: a stylegan encoder for image-to-image translation », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2287–2296.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala, « Unsupervised representation learning with deep convolutional generative adversarial networks », *in*: *arXiv preprint arXiv:1511.06434* (2015).

[Rob+19] Joseph P Robinson, Yuncheng Li, Ning Zhang, Yun Fu, and Sergey Tulyakov, « Laplace landmark localization », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10103–10112.

[Ros58] Frank Rosenblatt, « The perceptron: a probabilistic model for information storage and organization in the brain. », *in*: *Psychological review* 65.*6* (1958), p. 386.

[RR08] Christian Raymond and Giuseppe Riccardi, « Learning with noisy supervision for Spoken Language Understanding », *in*: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 4989–4992, DOI: 10.1109/ICASSP.2008.4518778.

[Rua+21] Zeyu Ruan, Changqing Zou, Longhai Wu, Gangshan Wu, and Limin Wang, « SADRNet: Self-aligned dual face regression networks for robust 3D dense face alignment and reconstruction », *in*: *IEEE Transactions on Image Processing* 30 (2021), pp. 5793–5806.

[RV05] S. Romdhani and T. Vetter, « Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior », *in*: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, 986–993 vol. 2, DOI: 10.1109/CVPR.2005.145.

[Sag+13] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic, « 300 faces in-the-wild challenge: The first facial landmark localization challenge », *in*: *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 397–403.

[San+19] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black, « Learning to regress 3D face shape and expression from an image without 3D supervision », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7763–7772.

[Sha+20] Taizhang Shang, Qiuju Dai, Shengchen Zhu, Tong Yang, and Yandong Guo, « Perceptual extreme super-resolution network with receptive field block », *in*: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 440–441.

[Shu+19] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng, « Meta-weight-net: Learning an explicit mapping for sample weighting », *in*: *Advances in neural information processing systems* 32 (2019).

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, « Dropout: a simple way to prevent neural networks from overfitting », *in*: *The journal of machine learning research* 15.*1* (2014), pp. 1929–1958.

[SS17] Ozan Sener and Silvio Savarese, « Active learning for convolutional neural networks: A core-set approach », *in*: *arXiv preprint arXiv:1708.00489* (2017).

[SS18] Ozan Sener and Silvio Savarese, « Active Learning for Convolutional Neural Networks: A Core-Set Approach », *in*: *International Conference on Learning Representations*, 2018, URL: https://openreview.net/forum?id=H1aIuk-RW.

[SSG22] Axel Sauer, Katja Schwarz, and Andreas Geiger, « Stylegan-xl: Scaling stylegan to large diverse datasets », *in*: *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.

[Str74] Wolfgang Straßer, « Schnelle kurven-und flächendarstellung auf grafischen sichtgeräten », PhD thesis, 1974.

[Sun+19] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang, « High-Resolution Representations for Labeling Pixels and Regions », *in*: *CoRR* abs/1904.04514 (2019).

[Tew+17] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt, « Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction », *in*: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1274–1283.

[Thi+16] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner, « Face2face: Real-time face capture and reenactment of rgb videos », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.

[Tov+21] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or, « Designing an encoder for stylegan image manipulation », *in*: *ACM Transactions on Graphics (TOG)* 40.*4* (2021), pp. 1–14.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, « Attention is all you need », *in*: *Advances in neural information processing systems* 30 (2017).

[Wan+22] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen, « High-fidelity gan inversion for image attribute editing », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11379–11388.

[WBF19] Xinyao Wang, Liefeng Bo, and Li Fuxin, « Adaptive wing loss for robust face alignment via heatmap regression », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6971–6981.

[Woo+22] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Matthew Johnson, Jingjing Shen, Nikola Milosavljević, Daniel Wilde, Stephan Garbin, Toby Sharp, Ivan Stojiljković, Tom Cashman, and Julien Valentin, « 3D Face Reconstruction with Dense Landmarks », *in*: *Computer Vision – ECCV 2022*, ed. by Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, Cham: Springer Nature Switzerland, 2022, pp. 160–177, ISBN: 978-3-031-19778-9.

[Wu+18] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou, « Look at boundary: A boundary-aware face alignment algorithm », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2129–2138.

[WXN21] Cho-Ying Wu, Qiangeng Xu, and Ulrich Neumann, « Synergy between 3dmm and 3d landmarks for accurate 3d facial geometry », *in*: *2021 International Conference on 3D Vision (3DV)*, IEEE, 2021, pp. 453–463.

[XD13] Xuehan Xiong and Fernando De la Torre, « Supervised descent method and its applications to face alignment », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 532–539.

[Xia+18] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun, « Unified perceptual parsing for scene understanding », *in*: *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 418–434.

[Xia+22] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang, « Gan inversion: A survey », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[Xu+21] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou, « Generative hierarchical features from synthesizing images », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4432–4442.

[Yan+13] Junjie Yan, Zhen Lei, Dong Yi, and Stan Li, « Learn to combine multiple hypotheses for accurate face alignment », *in*: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 392–396.

[Yao+22] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier, « A Style-Based GAN Encoder for High Fidelity Reconstruction of Images and Videos », *in*: *European conference on computer vision* (2022).

[YK19] Donggeun Yoo and In So Kweon, « Learning loss for active learning », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 93–102.

[ZBT22] Wojciech Zielonka, Timo Bolkart, and Justus Thies, « Towards metrical reconstruction of human faces », *in*: *European Conference on Computer Vision*, Springer, 2022, pp. 250–269.

[Zha+17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, « Understanding deep learning requires rethinking generalization », *in*: *International Conference on Learning Representations*, 2017, URL: `https://openreview.net/forum?id=Sy8gdB9xx`.

[Zha+21] Long Zhao, Zizhao Zhang, Ting Chen, Dimitris Metaxas, and Han Zhang, « Improved transformer for high-resolution gans », *in*: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18367–18380.

[Zha+22] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo, « Styleswin: Transformer-based gan for high-resolution image generation », *in*: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11304–11314.

[Zhe+22] Yinglin Zheng, Hao Yang, Ting Zhang, Jianmin Bao, Dongdong Chen, Yangyu Huang, Lu Yuan, Dong Chen, Ming Zeng, and Fang Wen, « General Facial Representation Learning in a Visual-Linguistic Manner », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18697–18709.

[Zho+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, « Learning deep features for discriminative localization », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[Zhu+16] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li, « Face alignment across large poses: A 3d solution », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 146–155.

[Zhu+17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, « Unpaired image-to-image translation using cycle-consistent adversarial networks », *in*: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[Zhu+19] Meilu Zhu, Daming Shi, Mingjie Zheng, and Muhammad Sadiq, « Robust facial landmark detection via occlusion-adaptive deep networks », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3486–3496.

[ZIE16] Richard Zhang, Phillip Isola, and Alexei A Efros, « Colorful image colorization », *in*: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, Springer, 2016, pp. 649–666.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Titre : Apprentissage par transfert pour l'analyse faciale avec des données annotées limitées et incohérentes

Mot clés : apprentissage profond, apprentissage par transfert, apprentissage actif, vision par ordinateur, alignement facial, reconstruction 3D de visage

Résumé : L'apprentissage profond s'est grandement développé ces dernières années. Cependant, beaucoup des méthodes existantes reposent toujours sur l'apprentissage supervisé qui requiert des données annotées. Or, obtenir ce type de données peut se révéler difficile. Dans ce mémoire, nous présentons une méthodologie, basée sur l'apprentissage par transfert, pour l'entraînement de réseaux de neurones avec un faible volume de données annotées. Notre approche consiste à augmenter avec de nouvelles couches et connexions un réseau génératif autosupervisé pré-entraîné, pour l'adapter à une tâche image-vers-image supervisée. Contrairement à la plupart des méthodes basées sur l'apprentissage par transfert, nous utilisons l'ensemble du modèle génératif, notamment le décodeur, pour la tâche supervisée. Notre méthodologie s'inspire du réseau 3FabRec proposé par Browatzki et al. pour l'alignement facial que nous avons étendu à différentes tâches supervisées et réseaux génératifs. Nous avons également proposé et étudié différentes façons d'augmenter le réseau génératif pour la tâche supervisée. Nous avons appliqué notre méthodologie à deux tâches supervisées : l'alignement facial et la reconstruction 3D de visage. Pour la première application, nos modèles ont dépassé l'état de l'art sur de nombreux jeux de données quand le nombre de données d'entraînement est limité. Pour la reconstruction 3D de visage, nous avons pu améliorer les prédictions d'un réseau autosupervisé via l'ajout d'information supervisée mais obtenue avec très peu de données annotées.

Title: Transfer learning for facial analysis with limited and inconsistent annotations

Keywords: deep learning, transfer learning, active learning, computer vision, face alignment, 3D face reconstruction

Abstract: Deep learning has developed considerably in recent years. However, many existing methods are still based on supervised learning, which requires annotated data. Obtaining such data can be difficult. In this thesis, we present a methodology, based on transfer learning, for training neural networks with a low volume of annotated data. Our approach consists in augmenting a pre-trained self-supervised generative network with new layers and connections, to adapt it to a supervised image-to-image task. Unlike most methods based on transfer learning, we use the entire generative model, including the decoder, for the supervised task. Our methodology is inspired by the 3FabRec network proposed by Browatzki et al. for face alignment, which we have extended to other supervised tasks and generative networks. We have also proposed and studied different ways of augmenting the generative network for the supervised task. We applied our methodology to two supervised tasks: face alignment and 3D face reconstruction. For the first application, our models outperformed the state-of-the-art on many datasets when the number of training data is limited. For 3D face reconstruction, we were able to improve the predictions of a self-supervised network via the addition of supervised information, but obtained with very little annotated data.