NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

BSc THESIS

Facial Inpainting Methods for Robust Face Recognition

Vasileios-Marios P. Panagakis

Supervisor: **Yannis Panagakis,** Associate Professor

ATHENS

OCTOBER 2021

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μέθοδοι Ανάπλασης Προσώπου για Εύρωστη Αναγνώριση Προσώπου

Βασίλειος-Μάριος Π. Παναγάκης

Επιβλέπων: **Γιάννης Παναγάκης,** Αναπληρωτής Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2021

BSc THESIS

Facial Inpainting Methods for Robust Face Recognition

Vasileios-Marios P. Panagakis
S.N.: 1115201600123

SUPERVISOR: **Yannis Panagakis,** Associate Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μέθοδοι Ανάπλασης Προσώπου για Εύρωστη Αναγνώριση Προσώπου

Βασίλειος-Μάριος Π. Παναγάκης
Α.Μ.: 1115201600123

ΕΠΙΒΛΕΠΩΝ: Γιάννης Παναγάκης, Αναπληρωτής Καθηγητής

ABSTRACT

Human face is probably the most characteristic identifier in every aspect of a person's life. In modern times, the development of cameras and digital electronics, has led to a non-stop generation and collection of face images enabling applications in numerous fields, like education, health, gaming, security, criminal and forensic investigation. It's obvious, that the progress in these fields can facilitate people's daily life and help them live in more secure societies. In order for this kind of applications to function properly, though, faces of high clearness and sharpness are required to be captured. This request is far from easy to satisfy in real world conditions. Occlusions such as eyeglasses, sunglasses, face masks, scarves, hands and more, cause serious corruptions to the face images and weaken the identification performance of face-related applications.

Although some algorithms can handle face recognition with occlusion, they still suffer from performance degradation due to occlusion's extent. Therefore, the removal of occlusions in face images is a very important, yet challenging task. The difficulty of the task lies in the fact that, a reconstruction method has to find a way to restore the occluded face parts to a non-occluded form, aiming to the generation of a clean face. As we know, human faces have similar shapes and appearances in general. However, the feature details may differ substantially among people depending on their race, gender and age. These details are the ones that raise even more the degree of difficulty of the face restoration procedure.

The objective of this thesis is the restoration of occluded face images to a non-occluded form, in order to facilitate their identification. To achieve that, we investigate a number of inpainting models and we evaluate them on face recognition task. The models are based on two principal face inpainting methodologies. The first, supervised method, known as Generative Landmark Guided Face Inpainting (or LaFIn) [17] exploits some of the most innovative and state-of-the-art tools, in the machine learning field, the deep neural networks. LaFIn's architecture benefits from the integration of facial landmarks and accomplishes the desired face restoration. The second, unsupervised method known as Principal Component Pursuit using Side Information, Features and Missing Values (or PCPSFM) [21] is a variation of the famous Robust Principal Component Analysis (RPCA) method. PCPSFM utilizes domain dependent prior knowledge and manages to recover a low-rank matrix L_0, containing the inpainted face. At the same time, it isolates the occlusions in a separate, sparse matrix S_0.

To evaluate the proposed methods, we worked on a portion of the popular CelebA dataset, which contains face representations of numerous celebrities. For the purpose of our experiments, we created occlusions of different sizes and shapes, in order to test the models under multiple scenarios. Concerning the evaluation process, three different models were employed to detect the dominant inpainting method, based on the percentage of successful matches between the inpainted faces and the clean faces of all the celebrity identities in the dataset.

SUBJECT AREA: Image Processing, Computer Vision, Deep Learning

KEYWORDS: Image Inpainting, Face Occlusions, Machine Learning, Neural Networks, Robust Principal Component Analysis, Face Recognition

ΠΕΡΙΛΗΨΗ

Το ανθρώπινο πρόσωπο είναι πιθανώς το πιο χαρακτηριστικό αναγνωριστικό της ταυτότητας ενός ανθρώπου σε κάθε έκφανση της ζωής του. Στη σύχρονη εποχή, η ανάπτυξη των καμερών και των ηλεκτρονικών συσκευών έχει οδηγήσει στην αδιάκοπη παραγωγή και συλλογή εικόνων με πρόσωπα, που βρίσκουν εφαρμογή σε πολλούς τομείς, όπως η εκπαίδευση, η υγεία, τα ηλεκτρονικά παιχνίδια, η ασφάλεια, η ποινική και ιατροδικαστική έρευνα. Είναι προφανές, ότι η πρόοδος σε αυτούς τους τομείς μπορεί να διευκολύνει την καθημερινή ζωή των ανθρώπων και να τους βηθήσει να ζουν σε πιο ασφαλείς κοινωνίες. Όμως, για να μπορέσουν αυτού του είδους οι εφαρμογές να λειτουργήσουν ορθά, απαιτείται η φωτογραφική λήψη προσώπων μεγάλης καθαρότητας και ευκρίνειας. Αυτή η απαίτηση είναι κάτι παραπάνω από δύσκολο να ικανοποιηθεί στις πραγματικές συνθήκες διαβίωσης. Occlusions όπως γυαλιά μυωπίας, γυαλιά ηλίου, μάσκες προσώπου, φουλάρια, χέρια κ.ά. προκαλούν σοβαρές αλλοιώσεις στις φωτογραφίες με πρόσωπα και αποδυναμώνουν την απόδοση της ταυτοποίησης προσώπου, από τις αντίστοιχες εφαρμογές.

Παρόλο που ορισμένοι αλγόριθμοι μπορούν να διαχειριστούν την αναγνώριση προσώπου με occlusion, εξακολουθούν να υφίστανται μείωση στην απόδοσή τους εξαιτίας της έκτασης του occlusion. Επομένως, η αφαίρεση των occlusions από τις εικόνες με πρόσωπα είναι μια πολύ σημαντική, αλλά και απαιτητική εργασία. Η δυσκολία της οφείλεται στο γεγονός, ότι μια μέθοδος ανακατασκευής πρέπει να βρει κάποιον τρόπο, ώστε να αποκαταστήσει τα occluded μέρη του προσώπου σε μια μη occluded μορφή, στοχεύοντας στην παραγωγή ενός καθαρού προσώπου. Όπως γνωρίζουμε, τα ανθρώπινα πρόσωπα έχουν παρόμοιο σχήμα και μέγεθος σε γενικές γραμμές. Ωστόσο, ορισμένα χαρακτηριστικά μπορεί να διαφέρουν πολύ με βάση την φυλή, το γένος και την ηλικία τους. Αυτές οι λεπτομέρειες αυξάνουν ακόμα περισσότερο το βαθμό δυσκολίας της διαδικασίας αποκατάστασης του προσώπου.

Ο σκοπός αυτής της Πτυχιακής Μελέτης είναι η αποκατάσταση occluded εικόνων με πρόσωπα σε μια μη occluded μορφή, ώστε να διευκολυνθεί η ταυτοποίησή τους. Για να το πετύχουμε αυτό, διερευνούμε ένα πλήθος από μοντέλα, ειδικευμένα στην ανάπλαση του προσώπου και τα αξιολογούμε με βάση την απόδοσή τους στην αναγνώριση προσώπου. Τα μοντέλα στηρίζονται σε δύο κυρίαρχες μεθοδολογίες της ανάπλασης προσώπου. Η πρώτη, επιτηρούμενη μεθοδολογία, γνωστή ως Generative Landmark Guided Face Inpainting (ή LaFIn) [17] αξιοποιεί μερικά από τα πιο καινοτόμα και υπερσύγχρονα εργαλεία στο πεδίο της μηχανικής μάθησης, τα βαθειά νευρωνικά δίκτυα. Η αρχιτεκτονική του LaFIn επωφελείται από την ενσωμάτωση των διακριτών σημείων του προσώπου και επιτυγχάνει την επιθυμητή αποκατάστασή του. Η δεύτερη, μη επιτηρούμενη μέθοδος γνωστή ως Principal Component Pursuit using Side Information, Features and Missing Values (ή PCPSFM) [21] είναι μια γενίκευση της διάσημης μεθόδου Robust Principal Component Analysis (RPCA). Η PCPSFM αξιοποιεί την προϋπάρχουσα γνώση και καταφέρνει να ανακτήσει έναν πίνακα L_0, χαμηλού βαθμού, ο οποίος περιέχει το αναπλασμένο πρόσωπο. Ταυτόχρονα, απομονώνει τα occlusions σε έναν ξεχωριστό, αραιό πίνακα S_0.

Για να αξιολογήσουμε τις προτεινόμενες μεθόδους, δουλέψαμε σε ένα τμήμα του δημοφιλούς συνόλου δεδομένων CelebA, το οποίο περιέχει τις αναπαραστάσεις των προσώπων διάφορων διάσημων προσωπικοτήτων. Για τα πειράματά μας, δημιουργήσαμε occlusions διαφορετικών μεγεθών και σχημάτων, ώστε να αξιολογήσουμε τα μοντέλα υπό πολλαπλές

συνθήκες. Όσον αφορά την διαδικασία αξιολόγησης, χρησιμοποιήθηκαν τρία διαφορετικά μοντέλα, που προσπαθούν να εντοπίσουν την κυρίαρχη μεθόδο ανάπλασης, με βάση το ποσοστό των επιτυχημένων ταιριασμάτων μεταξύ των αναπλασμένων και των καθαρών προσώπων όλων των διάσημων προσωπικοτήτων, που εμπεριέχονται στο σύνολο δεδομένων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Εικόνας, Υπολογιστική Όραση, Βαθιά Μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Ανάπλαση Εικόνας, Εμφράξεις Προσώπου, Μηχανική Μάθηση, Νευρωνικά Δίκτυα, Εύρωστη Ανάλυση Κυρίων Συνιστωσών, Αναγνώριση Προσώπου

To my family.

ACKNOWLEDGEMENTS

For the present Bachelor's thesis, I would primarily like to thank my supervisor, professor Yannis Panagakis, for his guidance and advice that contributed towards successfully completing the whole project. I undertook a part of this project during my summer Internship in the "LIBRA A.I Technologies" enterprise. Hence, I would also like to express my gratitude to Mr. Athanasios Mpalomenos and Mr. Yannis Kopsinis for giving me the opportunity to join and evolve through a modern data science team.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

The thesis at hand was undertaken as part of the course of study for the undergraduate degree at the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. The relevant work was conducted from May 2021 to October 2021 in Athens under the supervision of professor Yannis Panagakis. The project was developed on a Linux machine using the Python programming language. For the development of the project, it was of great importance to get familiar with PyTorch and Tensorflow frameworks used to implement methods connected to the deep learning field, as well as with libraries related to face detection, recognition and identification.

1. INTRODUCTION

1.1 Motivation

Human face is probably the most characteristic identifier in every aspect of a person's life. In modern times, the development of cameras and digital electronics, has led to a non-stop generation and collection of face images enabling applications in numerous fields, like education, health, gaming, security, criminal and forensic investigation. For these activities to be functional complex face recognition systems have been built. Face recognition [18] is a method of identifying or verifying the identity of an individual using their face. Thus, these systems are used to identify people in photos, video, or in real-time. Face recognition systems use computer algorithms to pick out specific, distinctive details about a person's face. These details, such as distance between the eyes or shape of the chin, are then converted into a mathematical representation and compared to data on other faces collected in a face recognition database. As expected, face recognition systems vary in their ability to identify people under challenging conditions such as poor lighting, low quality image resolution, and suboptimal angle of view.

However, external conditions are not the only ones that may affect the quality of the face recognition process. Especially, in a real world scenario, occlusions such as eyeglasses, sunglasses, face masks, scarves, hands and more often hide big parts of human faces causing serious corruptions to face images and making the face recognition task particularly challenging. Although some algorithms can handle face recognition with occlusion, they still suffer from performance degradation due to occlusion's extent. Therefore, the removal of occlusions in face images is a very important, yet challenging task. The difficulty of the task lies in the fact that, a reconstruction method has to find a way to restore the occluded face parts to a non-occluded form, aiming to the generation of a clean face. As we know, human faces have similar shapes and appearances in general. However, the feature details may differ substantially among people depending on their race, gender and age. These details are the ones that raise even more the degree of difficulty of the face restoration procedure.

Figure 1.1: Real world occlusions [1]

1.2 Related Work

Various image inpainting methods have been developed over the last decades. In the following subsections, the most significant works are reviewed, split in two categories, the supervised and the unsupervised ones.

1.2.1 Supervised Methods

Deep learning-based methods is a major group of supervised methods, which deal with the image inpainting problem. The context encoder [5], which is treated as a pioneer deep-learning method for image completion, introduces an encoder-decoder network trained with an adversarial loss [6]. After that, plenty of follow-ups have been proposed to improve the performance from various aspects. For instance, the scheme in [7] employs both the global and local discriminators to accomplish the task. Another attempt suggested in [8] designs a coarse-to-fine network structure and applies a self-attention layer to connect related features at distant spatial locations. Besides, in [9] and [10] the convolutional layers are upgraded for making networks adaptive to the masked input. However, most of the aforementioned methods can barely keep the structure of the original image and the inpainted result frequently tends to be blurry, especially on large occluded areas. For the sake of maintaining the structure of corrupted images, a number of methods, such as [11, 12], try to firstly predict the edge information for corrupted images and then apply it as a condition to guide the inpainting. Even these methods are unable to predict reas-

[1] www.comp.nus.edu.sg/~leowwk/thesis/liguodong.pdf

onable edges inside the masked regions, due to the face deformation caused from the corruption.

Deep face inpainting methods is another significant group of supervised methods. Specific to face completion, the authors of [13] construct a loss, which takes care of the gap in semantic segmentation (face parsing), between the inpainted face images and the ground truth ones, expecting to achieve a better preservation of the face structure. However, this work often suffers from colour inconsistency and is unable to process effectively faces with large poses. [9, 14] suggest directly, that users should manually label face edges to get more accurate results. Although, this can be a flexible way to edit faces, sometimes it is difficult for users to input precise edge information. That's why, in [11] a network that predicts the edges is built, which however suffers from inaccurate prediction on large holes. Moreover, it seems that, for face completion, both face parsing and edge information are relatively redundant, which may even degenerate the performance when feeding slightly inaccurate information into the inpainting module. Facial landmarks are better to act as guidance, thanks to their neatness, sufficiency, and robustness to reflect the structure of face. Many works, such as [15, 16] have successfully applied landmarks to the task of face generation.

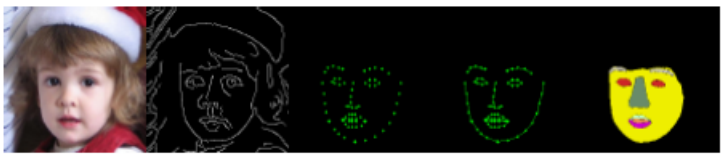

Figure 1.2: An illustration of different facial features. From left to right: the input, Canny edges, landmarks, edges by connecting the landmarks, parsing regions [17]

1.2.2 Unsuvervised Methods

Concerning the unsupervised inpainting methods, the traditional methods of the category, consist of two main representative branches, the diffusion-based and patch-based approaches. Diffusion-based approaches [1, 2] propagate low-level features around the occluded areas, in an iterative way. However, these methods are only effective in reforming regions without structure. At the same time, patch-based methods [3, 4] attempt to copy similar blocks from either the same image or a set of images to the target regions. The computational cost of calculating the similarity between their blocks is expensive, even though some works like [3] have been proposed, in order to accelerate the procedure. On the other hand, as a common limitation, they all hypothesize that the missing part can be found elsewhere, which does not always hold in practice.

Another group of unsupervised inpainting methods is called the non-blind inpainting methods. Those techniques fill in the occluded part of an image using the pixels around the missing region. Exemplar-based techniques that cheaply and effectively generate new texture by sampling and copying color values from the source are widely used. In paper [19], a non-blind inpainting method suggests a unified scheme to determine the fill order of the target region, using an exemplar-based texture synthesis technique. The confidence value of each pixel and image isophotes are combined to determine the priority of filling. [20] presents an image inpainting technique to remove occluded pixels when the occlusion is small. More specifically, it combines feature extraction and fast weighted principal component analysis (FW-PCA) to restore the occluded images.

1.3 Objective

The objective of this thesis is the restoration of occluded face images to a non-occluded form, in order to facilitate their identification. To achieve that, we investigate a number of inpainting models and we evaluate them on face recognition task. The models are based on two principal face inpainting methodologies. The first, supervised method, known as Generative Landmark Guided Face Inpainting (or LaFIn) exploits some of the most innovative and state-of-the-art tools, in the machine learning field, the deep neural networks. LaFIn's architecture benefits from the integration of facial landmarks and accomplishes the desired face restoration. The second, unsupervised method known as Principal Component Pursuit using Side Information, Features and Missing Values (or PCPSFM) is a variation of the famous Robust Principal Component Analysis (RPCA) method. PCPSFM utilizes domain dependent prior knowledge and manages to recover a low-rank matrix L_0, containing the inpainted face. At the same time, it isolates the occlusions in a separate, sparse matrix S_0.

To evaluate the proposed methods, we worked on a portion of the popular CelebA dataset, which contains face representations of numerous celebrities. For the purpose of our experiments, we created occlusions of different sizes and shapes, in order to test the models under different scenarios. We will elaborate more on the dataset structure and the types of occlusions in the upcoming chapters of the thesis.Concerning the evaluation process, three different evaluation models were used aiming to detect the dominant inpainting method, based on the percentage of successful matches between the inpainted faces and the clean faces of all the celebrity identities in the dataset.

2. LAFIN: GENERATIVE LANDMARK GUIDED FACE INPAINTING

Face restoration has proven to be a particularly challenging task. Hence, a qualified face inpainting algorithm should take into account the following two conditions to guarantee realistic outputs:

- **Face structure:** The topological relationship among facial features including eyebrows, eyes, nose and mouth must always be well-organized. All the inpainted faces must satisfy this topology structure.
- **Consistent face attributes:** Attributes such as pose, gender, ethnicity, and expression, should be consistent across the inpainted face parts and the non-occluded areas.

Generative Landmark Guided Face Inpaintor (LaFIn) [17], is a deep network built to carry out the face inpainting problem. LaFIn comprises of a Landmark Prediction Module and an Image Inpainting Module, trained on the CelebA and Celeba-HQ datasets [28].

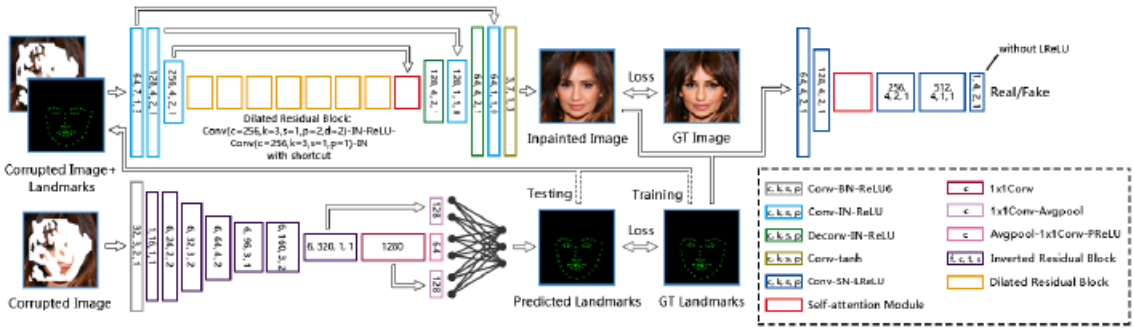

Figure 2.1: LaFIn architecture. At first, the Landmark prediction module estimates the landmarks and then the Inpainting module applies them on the corrupted image [17]

The functionality of LaFIn can be summarized as follows:

1. Building of a module that predicts landmarks on incomplete faces. The landmarks reflect the topological structure, pose and expression of the target face.
2. Implementation of an image inpainting module that employs the predicted landmarks as guidance, in order to accomplish the face restoration. To achieve attribute consistency, the module utilizes distant spatial context and connects temporal feature maps.

2.1 Why adopt landmarks?

As stated, LaFIn uses landmarks as guidance to detect the facial regions of the corrupted faces. Landmarks can be viewed as the discrete points sampled on the key regions of a face. Their strong attribute is that they are able to reform the key facial regions without containing redundant information. This attribute makes them compact, sufficient, and robust.

Yet, someone may wonder why use landmarks as guidance, instead of edge or parsing information. Indeed, edge or parsing information can be highly accurate when clean faces are studied. But this is not the case in challenging situations, where faces with multiple and large corruptions have to be analysed. In this type of situations, it is impossible to generate reasonable edges. As a result, the edge information retrieved would be redundant and inaccurate, contributing to the creating of an underperforming inpainting algorithm.

On the other hand, a set of landmarks is always available, no matter what situation the face is in. Once the landmarks are obtained, they immediately determine the topology structure, pose and expression of the face, as shown in Figures 2.2, 2.3. For the reasons above, including the fact they are much more convenient to control from an editing perspective, we reach the conclusion that using landmarks as guidance is the best available choice to achieve the face restoration.

Figure 2.2: Landmarks on clean faces

Figure 2.3: Landmarks on occluded faces

2.2 How to guarantee attribute consistency?

Except for the pose and expression attributes determined by the landmarks, there are several other attributes, such as gender, age, ethnicity, needed to be taken into consideration. By the term consistency we mean the necessity to connect the clean and the inpainted parts of the face in a smooth way, so that there is no visual differentiation between them. To fulfill the consistency requirement, the inpainting algorithm should take the information of the clean parts as a reference point to reconstruct the occluded parts.

In practice, this can be achieved by connecting distant spatial context and temporal feature maps. Specifically, having a larger receptive field allows a network to capture more spatial context. In the context of Single Image Super Resolution (SISR), this increases the ability of the network to reconstruct larger and more complex edge structures. Respectively, the use of Long short-term memory (LSTM) [32] neural networks, allows to keep track of long-term temporal dependencies that determine the variability of features.

LSTMs are artificial Recurrent Neural Networks (RNN) built in a way that makes them suitable to maintain information in their memory cells for long periods of time.

2.3 Network Architecture

LaFIn is a neural network, composed of two main subnets. The first, Landmark Prediction Module, is the one that predicts the landmark locations, while the second, Image Inpainting Module, generates new pixels conditioned on the predicted landmarks, as illustrated in Figure 2.1. In the following subsections, we will break down LaFIn's architecture in detail.

2.3.1 Landmark Prediction Module

The goal of the Landmark Prediction Module is to retrieve a set of 68 landmarks from a corrupted face image. For the purposes of the face inpainting task, we are more concerned about getting landmarks that can accurately identify the face structure and its basic attributes, like pose and expression, rather than finding the precise location of each unique face landmark. The reason behind this simplification is that most of the in-between landmarks don't offer important information, concerning the face inpainting task.

LaFIn's Landmark Prediction Module follows the same architecture as most of the preexistent landmark detectors (see [22, 23]). Specifically, it is built upon the MobileNet-V2 model, proposed in "Mobilenetv2: Inverted residuals and linear bottlenecks" [24] and focuses on feature extraction.

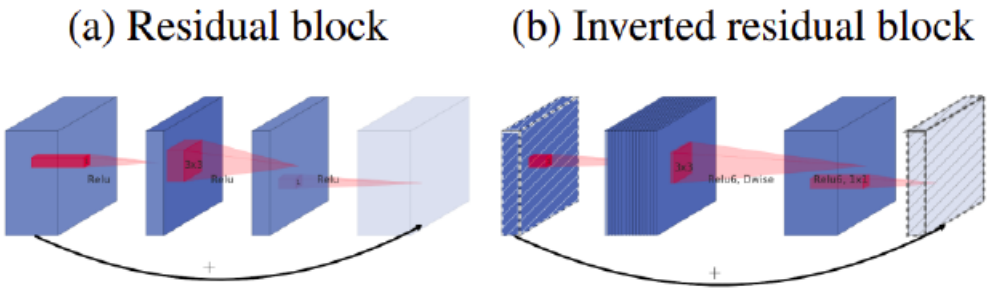

Figure 2.4: Evolution of residual blocks. (a) MobileNets: Residual block, (b) MobileNet-V2: Inverted residual block [24]

Basically, [24] improved the previous work of MobileNets [25], by introducing novel inverted residuals and linear bottlenecks (Figure 2.4b). To achieve that, the authors of [24] optimized the preexistent residual blocks [26] (Figure 2.4a) by establishing a series of novelties, concerning the ReLU activation function and its transformation. Moreover, instead of compressing the input feature map and connecting the layers with a high number of channels, they chose to expand the input map and used shortcuts directly between the bottlenecks, since they contain all the information. Additionally, they used linear activation functions to set up the input bottleneck to prevent non-linearities from erasing much information.

The process that takes places between the three layers of each residual block, can be briefly described as follows:

1. The first layer decompresses the data to its original form.
2. A depthwise layer replaces the typical convolution layer and performs filtering using the ReLU functions.
3. The last layer restores the data to its compact form.

This way, MobileNet-V2 deals with the loss of information issue, caused in the classical residual blocks of MobileNets due to their inability to filter a high-dimensional tensors with ReLU. In fact, the upgraded MobileNet-V2 architecture leads to a 75% reduction of the number of network's parameters and increases the achieved mean average precision on the ImageNet dataset by 1.6%, relative to MobileNets.

This exact architecture is used in the implementation of LaFIn's Landmark Prediction Module. The final landmark prediction is achieved by fully connecting the fused feature maps at different rear stages, as shown in Figure 2.1.

2.3.2 Image Inpainting Module

The purpose of the Image Inpainting Module is to restore faces by taking occluded images and their predicted landmarks. LaFIn's authors use the famous Generative Adversarial Network (GAN) architecture, proposed in "Generative Adversarial Nets" [6] by I. Goodfellow et al, to build the inpainting module.

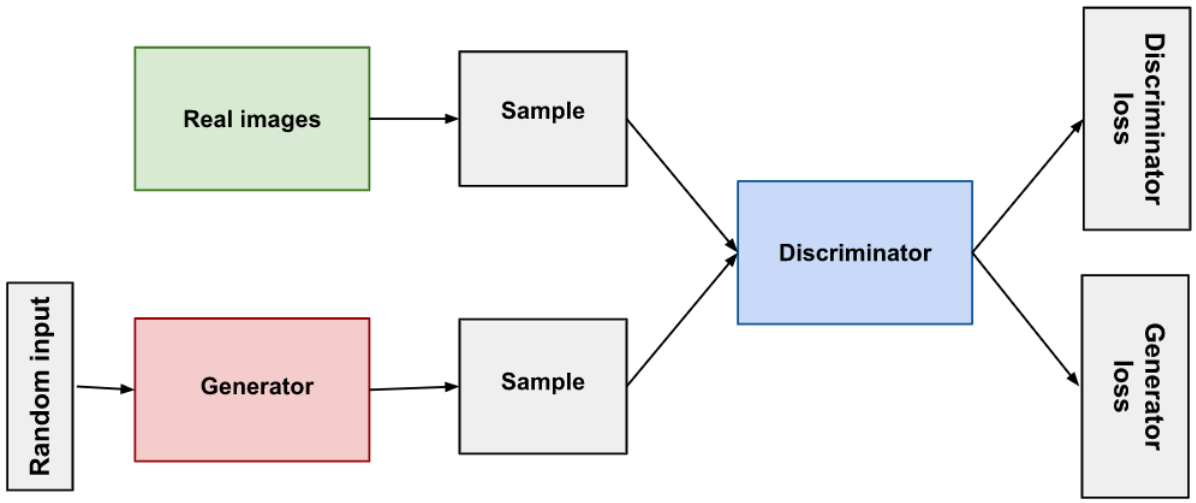

Figure 2.5: GAN schema [2]

GAN uses two networks, called the generator and the discriminator. Based on insights from game theory, GAN's training objective can be considered as a mini-max game, where the generator needs to produce fake, realistic images conditioned on the landmarks, out of a known prior distribution, in order to fool the discriminator. At the same time, the discriminator needs to distinguish between the real and the generated fake images. The networks are trained in a competitive adversarial manner. The convergence is reached when the generated results are not distinguishable from the real ones.

[2] https://developers.google.com/machine-learning/gan/gan_structure

Overall, LaFIn's generator is based on the U-Net structure [28]. U-Net is an architecture for semantic segmentation. It involves a contracting path and an expansive path. The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3×3 unpadded convolutions, each followed by a ReLU activation function and a 2×2 max pooling operation with stride 2 for downsampling. At each downsampling step the number of feature channels gets doubled. Every step in the expansive path consists of an upsampling of the feature map, followed by a 2×2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1×1 convolution is used to map each 64-component feature vector to the desired number of classes. In total, the network has 23 convolutional layers.

LaFIn's generator, in particular, consists of three gradually downsampled encoding blocks, followed by seven residual blocks with dilated convolutions and a long-short term attention block [29]. The latter connects temporal feature maps, while the stacked dilated blocks enlarge the receptive field, so that features located in a wider range can be taken into account. Afterwards, the decoder processes the feature maps gradually upsampled to the same size as the input. Besides, shortcuts are added between the corresponding encoder and decoder layers. It's worth mentioning that, the 1×1 convolution operation is executed before each decoding layer, so that the attention block can adjust the features weights retrieved from the corresponding previous layer. This specific architecture boosts the network's ability to utilize distant features in a both spatial and temporal manner.

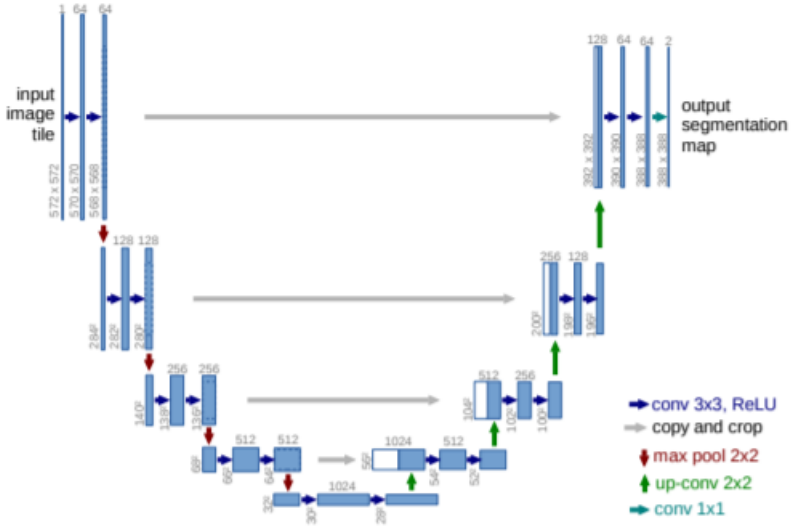

Figure 2.6: U-net architecture [28]

Concerning LaFIn's discriminator subnet, it is built upon the 70×70 PatchGAN architecture [30]. PatchGAN is a type of discriminator for GANs, which only penalizes structure at the scale of local image patches. This type of discriminator tries to decide, if each $N \times N$ ($N = 70$, in this work) patch in an image is real or fake. PatchGAN is run convolutionally across the image, averaging all responses to provide the final output. Such a discriminator models effectively the image as a Markov random field, assuming independence between pixels separated by more than a patch diameter. This procedure can be considered as a type of style loss.

Particularly, in LaFIn's discriminator blocks, the spectral normalization [31] technique is introduced aiming to stabilize the training process. Furthermore, an attention layer is inserted to focus on the attributes consistency. Finally, in contrast with works like [7], where two discriminators are deployed, (i.e., a global discriminator assesses an image's aggregate consistency, while a local one tries to ensure the local consistency of the inpainted region), LaFIn's inpainting module deploys just one discriminator, which only requires an image and its landmarks as input. This design choice leads to a lighter network architecture without affecting its performance because:

1. The global structure is guaranteed thanks to the landmarks conditioning.
2. The attribute consistency is ensured with the insertion of the attention layer.

2.4 FAN-Face

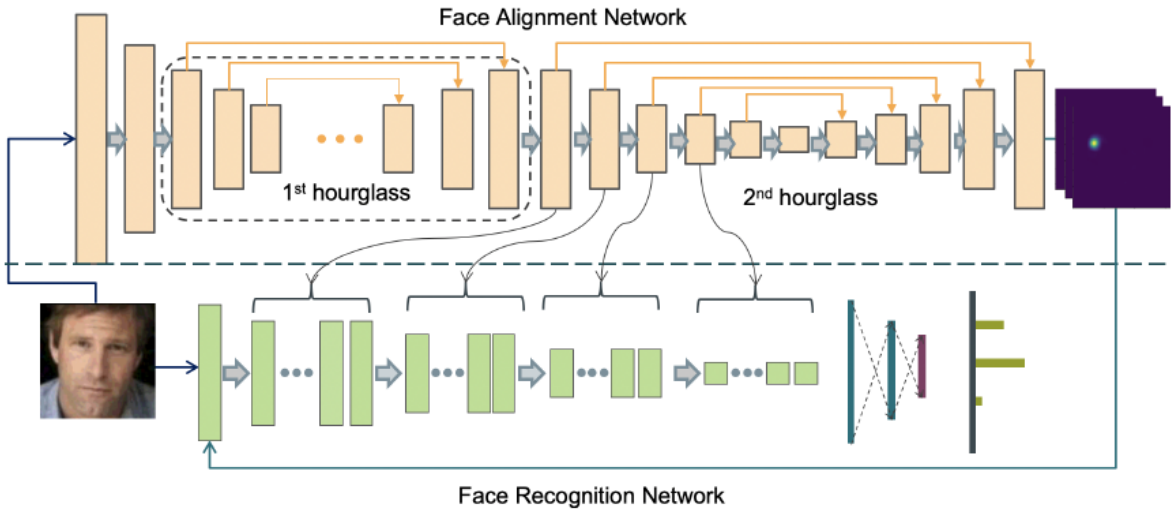

Figure 2.7: FAN-Face architecture [42]

LaFIn's Landmark Prediction Module doesn't always return a set of landmarks that can accurately identify the face structure and its basic attributes, specially in cases where the occlusions cover big parts of face images. Hence, LaFin's implementation embeds FAN-Face, another model suitable for landmark prediction, proposed in "FAN-Face: a Simple Orthogonal Improvement to Deep Face Recognition" [42].

FAN-Face is based on the integration of features from a facial landmark localization network and a face recognition network. The facial landmark localization network is a pre-trained Face Alignment Network (FAN) [44], which has been shown to robustly detect facial landmarks across large poses, facial expressions, illumination changes, low resolution and even occlusions. FAN is a stacked hourglass network [45] built using the residual block of [43]. After experimentation, the authors judged that 2 stacks suffice to achieve good accuracy. Face Recognition Network (FRN), is a ResNet [46], which is widely used in classification tasks and particularly in face recognition. The innovative idea behind Fan-Face is the integration of features from the pre-trained FAN into FRN, while training FRN. FRN is trained in standard ways on popular face recognition datasets, like VGGFace2, MS1MV2 and CASIA-Webface. The most significant features integrated from FAN are: the output in the form of facial landmark heatmaps and features from different layers extracted in different resolutions.

3. ROBUST PRINCIPAL COMPONENT ANALYSIS USING SIDE INFORMATION

3.1 Problem Definition

Suppose a data matrix M is given, which can be decomposed as

$$M = L_0 + S_0, \tag{3.1}$$

where L_0 is a low-rank matrix and S_0 is a sparse matrix. In this case, both components are of arbitrary magnitude. Neither the low-dimensional column and row space of L_0, nor their dimension is known. The locations of the non-zero entries of S_0 and their values are unknown, as well. The goal is to recover accurately or even exactly the low-rank and sparse components, in an efficient manner. In the case of the face inpainting problem, matrix M contains aligned face images, stacked as column vectors. The face images can be both occluded and non-occluded. The recovered L_0 matrix contains all the inpainted face images in the form of column vectors, while S_0 matrix consists of images, depicting the occluded parts of the initial faces, stacked in a corresponding format.

3.2 Principal Component Analysis

Principal Component Analysis (PCA) [33], is a dimensionality-reduction method that is often used to reduce the dimensionality of large datasets, by transforming a large set of variables into a smaller one that still contains most of the information. Reducing the number of variables of a dataset naturally comes at the expense of accuracy. Nevertheless, most of the times it is worth trading a little accuracy for more simplicity through the dimensionality reduction. PCA can be very effective when a big number of images has to be managed. Images are combinations of pixels in rows and columns, placed one after another. Each pixel represents a single image's intensity value. Therefore, to process multiple images we can form a matrix considering a row or column of pixels as a vector. It is obvious, that working with many images requires huge amounts of storage and processing power, whereas PCA compresses the whole image data, helping to preserve them in smaller storage facilities and analyze them in a much easier way.

Mathematically, we could say that PCA seeks the best (in an l^2 sense) rank-k estimate of L_0 by solving

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & ||M - L|| \\ \text{subject to} \quad & \text{rank}(L) \leq k. \end{aligned} \tag{3.2}$$

where $||M||$ denotes the 2-norm, which is the largest singular value of M. This problem can be efficiently solved via the Singular Value Decomposition (SVD) [34].

3.3 Robust Principal Component Analysis

Although, PCA is arguably the most widely used statistical tool for data analysis and dimensionality reduction, its application concerning grossly corrupted data seems to be pretty fragile. Basically, when applied in a single extremely corrupted entry in M it may render the estimated $\hat{\mathsf{L}}$ arbitrarily far from the true L_0 matrix. In fact, gross errors are now ubiquitous in modern data-centric applications, because some measurements may be arbitrarily corrupted, due to malicious tampering, sensor failures or face occlusions, in our case.

3.3.1 Problem Variation

A new problem, considered as an idealized version of RPCA set to replace the initial one aiming to deal with the case of huge corruptions. In the new problem the purpose is to recover a low-rank matrix L_0 from highly corrupted measurements

$$\mathsf{M} = \mathsf{L}_0 + \mathsf{S}_0, \tag{3.3}$$

The entries in S_0 can have arbitrarily large magnitude and their support is assumed to be sparse, but unknown. Even if, at first sight, the separation problem seems impossible to solve, since the unknowns to infer for L_0 and S_0 are twice as many as the given measurements in $\mathsf{M} \in \mathbb{R}^{n_1 \times n_2}$, E. Candes et al. showed in "Robust Principal Component Analysis?" [35], that not only this problem can be solved, but it can be solved by tractable convex optimization.

Let $||\mathsf{M}||_* \coloneqq \sum_i \sigma_i(\mathsf{M})$ denote the nuclear norm of the matrix M, i.e. the sum of the singular values of M, and let $||\mathsf{M}||_1 = \sum_{ij} (|\mathsf{M}_{ij}|)$ denote the l_1-norm of M seen as a long vector in $\mathbb{R}^{n_1 \times n_2}$. The authors of [35] ended up showing that under rather weak assumptions, the Principal Component Pursuit (PCP) solves

$$\begin{aligned} \underset{\mathsf{L},\mathsf{S}}{\text{minimize}} \quad & ||\mathsf{L}||_* + \lambda||\mathsf{S}||_1 \\ \text{subject to} \quad & \mathsf{M} = \mathsf{L} + \mathsf{S}. \end{aligned} \tag{3.4}$$

and recovers exactly the low-rank L_0 and the sparse S_0. In fact, they proved, that the above problem can be solved by efficient and scalable algorithms, at a cost not so much higher than the classical PCA one.

3.3.2 Problem Solution

The authors chose to solve the convex PCP problem (3.4) using an Augmented Lagrange Multiplier (ALM) algorithm introduced in [36]. ALM has the ability to achieve high accuracy rates in a small number of iterations and it works stably across a wide range of problem settings, without needing parameter tuning. Actually, during the execution of their experiments they observed, that the number of iterations often remains bounded by rank(L_0) throughout the ALM optimization, which reinforces the efficient repetition of the precedure.

Let $\langle \mathsf{A}, \mathsf{B} \rangle$ represent $tr(\mathsf{A}^T\mathsf{B})$ for real matrices A, B. The ALM method operates on the augmented Lagrangian

$$l(\mathsf{L}, \mathsf{S}, \mathsf{Y}) = ||\mathsf{L}||_* + \lambda||\mathsf{S}||_1 + \langle \mathsf{Y}, \mathsf{M} - \mathsf{L} - \mathsf{S} \rangle + \frac{\mu}{2}||\mathsf{M} - \mathsf{L} - \mathsf{S}||_F{}^2. \tag{3.5}$$

A generic Lagrange multiplier algorithm [37] would solve the PCP problem by repeatedly setting $(\mathsf{L}_k, \mathsf{S}_k) = arg\,min_{\mathsf{L},\mathsf{S}}\, l(\mathsf{L}, \mathsf{S}, \mathsf{Y}_k)$, and then it would update the Lagrange multiplier matrix via $\mathsf{Y}_{k+1} = \mathsf{Y}_k + \mu(\mathsf{M} - \mathsf{L}_k - \mathsf{S}_k)$. For this particular PCP problem (3.4), there was no need in solving a sequence of convex programs, after recognizing that both $min_{\mathsf{L}}\, l(\mathsf{L}, \mathsf{S}, \mathsf{Y})$ and $min_{\mathsf{S}}\, l(\mathsf{L}, \mathsf{S}, \mathsf{Y})$ have very simple and efficient solutions. The authors let $S_\tau : \mathbb{R} \to \mathbb{R}$ denote the shrinkage operator $S_\tau(x) = sgn(x)\, max(|x| - \tau, 0)$, which naturally extends to matrices, $S_\tau(\mathsf{A})$ by applying it to matrix A element-wise. This way, they showed

$$arg\ \min_{\mathsf{S}}\ l(\mathsf{L}, \mathsf{S}, \mathsf{Y}) = S_{\lambda\mu}(\mathsf{M} - \mathsf{L} + \mu^{-1}\mathsf{Y}). \tag{3.6}$$

Similarly, they let $D_\tau(\mathsf{A})$ denote the singular value thresholding operator given by $D_\tau(\mathsf{A}) = \mathsf{U}S_\tau(\Sigma)\mathsf{V}^T$, where $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^T$ is the SVD of A. In a similar way, they showed

$$arg\ \min_{\mathsf{L}}\ l(\mathsf{L}, \mathsf{S}, \mathsf{Y}) = D_{\mu}(\mathsf{M} - \mathsf{S} - \mu^{-1}\mathsf{Y}). \tag{3.7}$$

Thus, they ended up solving a difficult convex problem, following a very practical strategy. The three most important steps of the strategy can be summarized as follows:

1. Minimize l with respect to L (fixing S).
2. Minimize l with respect to S (fixing L).
3. Update the Lagrange multiplier matrix Y based on the residual $\mathsf{M} - \mathsf{L} - \mathsf{S}$.

The above steps are represented in a mathematical way in the following Algorithm 1.

Algorithm 1 Principal Component Pursuit by Alternating Directions

Initialize: $\mathsf{S}_0 = \mathsf{Y}_0 = 0, \mu > 0$.
while not converged **do**
  $\mathsf{L}_{k+1} = D_{\mu}(\mathsf{M} - \mathsf{S}_k - \mu^{-1}\mathsf{Y}_k)$
  $\mathsf{S}_{k+1} = S_{\lambda\mu}(\mathsf{M} - \mathsf{L}_{k+1} + \mu^{-1}\mathsf{Y}_k)$
  $\mathsf{Y}_{k+1} = \mathsf{Y}_k + \mu(\mathsf{M} - \mathsf{L}_{k+1} - \mathsf{S}_{k+1})$
end while
Return: L, S.

Algorithm 1 is a special case of a more general class of augmented Lagrange multiplier algorithms, known as alternating directions methods [36]. Via their experiments, the authors proved that the algorithm requires a relatively small numbers of iterations to achieve good accuracy. The main computational cost of each iteration is due to the use of singular value thresholding to calculate L_{k+1}. The value of L_{k+1} depends on the calculation of the singular vectors of $\mathsf{M} - \mathsf{S}_k - \mu^{-1}\mathsf{Y}_k$, whose corresponding singular values exceed the threshold μ. In fact, they observed empirically, that the number of such large singular values is often bounded by rank(L_0), allowing the efficient calculation of each next iteration via a partial SVD, which gradually leads to a significant cost reduction. Last but not least, it is of a great importance to mention the implementation details for Algorithm 1 and particularly the value of μ and the stopping criterion. The authors use $\mu = n_1 n_2/4||\mathsf{M}||_1$, as suggested and explained in [36]. Respectively, the algorithm is terminated, when $||\mathsf{M} - \mathsf{L} - \mathsf{S}||_F \leq \delta||\mathsf{M}||_F$, with $\delta = 10^{-7}$.

3.4 Robust Principal Component Analysis using Side Information, Features and Missing Values

3.4.1 Problem Upgrade

Over the years, many variants have been proposed, trying to confront the convex PCP problem (3.4) in a more efficient way for a number of different applications, including background modelling from surveillance video and removing shadows or specularities from face images. Two important variants were presented in [38] and [39].

Principal Component Pursuit with Features (PCPF) method in [38] assumes that there are available orthogonal column spaces $\mathsf{U} \in \mathbb{R}^{n_1 \times d_1}$, where $d_1 \leq n_1$ and row spaces $\mathsf{V} \in \mathbb{R}^{n_2 \times d_2}$, where $d_2 \leq n_2$, with the following objective:

$$\begin{aligned} \underset{\mathsf{H},\mathsf{E}}{\text{minimize}} \quad & ||\mathsf{H}||_* + \lambda||\mathsf{E}||_1 \\ \text{subject to} \quad & \mathsf{X} = \mathsf{U}\mathsf{H}\mathsf{V}^T + \mathsf{E}, \end{aligned} \tag{3.8}$$

where $\mathsf{H} \in \mathbb{R}^{d_1 \times d_2}$ is a bilinear mapping for the recovered low-rank matrix $\mathsf{L} \in \mathbb{R}^{d_1 \times d_2}$, with rank $r \ll min(n_1, n_2)$ and $\mathsf{E} \in \mathbb{R}^{n_1 \times n_2}$ is a sparse matrix with entries of arbitrary magnitude. The main drawback of this model is that features need to be accurate and noiseless, which is not trivial in practical scenarios.

In the case of missing data, robust matrix recovery method [39] enhances PCP to deal with occlusions:

$$\begin{aligned} \underset{\mathsf{L},\mathsf{E}}{\text{minimize}} \quad & ||\mathsf{L}||_* + \lambda||\mathsf{W} \circ \mathsf{E}||_1 \\ \text{subject to} \quad & \mathsf{X} = \mathsf{L} + \mathsf{E}, \end{aligned} \tag{3.9}$$

where W is the matrix of binary occlusion masks and $\mathsf{A} \circ \mathsf{B}$ symbolises the element-wise multiplication of two matrices of the same dimension. The method's Jacobi-type update schemes can be implemented in parallel and hence are attractive for solving large-scale problems.

For the purposes of this thesis, we are going to experiment with the Robust Principal Component Pursuit using Side information, Features and Missing values (PCPSFM), proposed in "Side Information for Face Completion: A Robust PCA Approach" [21]. This work introduces a novel convex program to use side information, which is a noisy approximation of the low-rank component, within the PCP framework. Moreover, the suggested method is able to handle missing values, while the developed optimization algorithm grants better convergence rates. Last but not least, the introduced model is able to use side information to exploit prior knowledge regarding the column and row spaces of the low-rank component, expanding even more the potential of the algorithm.

At first, the authors of [21] presented a PCPSM model, which uses side information with missing values. In order for (3.10) to be valid, they set as a precondition, that a noisy estimate of the low-rank component of the data $\mathsf{S} \in \mathbb{R}^{n_1 \times n_2}$ must be available.

$$\begin{aligned} \underset{\mathsf{L},\mathsf{E}}{\text{minimize}} \quad & ||\mathsf{L}||_* + \alpha||\mathsf{L}-\mathsf{S}||_* + \lambda||\mathsf{W}\circ\mathsf{E}||_1 \\ \text{subject to} \quad & \mathsf{X} = \mathsf{L} + \mathsf{E}, \end{aligned} \tag{3.10}$$

where $\alpha > 0$, $\lambda > 0$ are parameters that weigh the effects of side information and noise sparsity.

Then, they utilized their proposed PCPSM model to generalise PCPF (3.8), which led to the introduction of the novel PCPSFM model, using side information, features and missing values.

$$\begin{aligned} \underset{\mathsf{H},\mathsf{E}}{\text{minimize}} \quad & ||\mathsf{H}||_* + \alpha||\mathsf{H}-\mathsf{D}||_* + \lambda||\mathsf{W}\circ\mathsf{E}||_1 \\ \text{subject to} \quad & \mathsf{X} = \mathsf{U}\mathsf{H}\mathsf{V}^T + \mathsf{E}, \quad \mathsf{D} = \mathsf{U}^T\mathsf{S}\mathsf{V}, \end{aligned} \tag{3.11}$$

where $\mathsf{H} \in \mathbb{R}^{d_1\times d_2}, \mathsf{D} \in \mathbb{R}^{d_1\times d_2}$ are bilinear mappings for the recovered low-rank matrix L and side information S respectively. The low-rank matrix L is recovered from the optimal solution $(\mathsf{H}^*, \mathsf{E}^*)$ to objective (3.11) via $\mathsf{L} = \mathsf{U}\mathsf{H}^*\mathsf{V}^T$.

3.4.2 Problem Solution

Similarly to the solution of the convex problem (3.4), the authors chose the multi-block Alternating Direction Method of Multipliers (ADMM) to deal with problem (3.11). As mentioned before, ADMM operates by carrying out repeated cycles of updates, until it converges. However, for ADMM to be effective it has to be made sure, that features U, V correspond to orthogonal matrices. Hence, (3.11) had to be transformed to the identical convex, but non-smooth problem:

$$\begin{aligned} \underset{\mathsf{H},\mathsf{E}}{\text{minimize}} \quad & ||\mathsf{H}||_* + \alpha||\mathsf{B}||_* + \lambda||\mathsf{W}\circ\mathsf{E}||_1 \\ \text{subject to} \quad & \mathsf{X} = \mathsf{U}\mathsf{H}\mathsf{V}^T + \mathsf{E}, \quad \mathsf{B} = \mathsf{H} - \mathsf{U}^T\mathsf{S}\mathsf{V}, \end{aligned} \tag{3.12}$$

where $\mathsf{H} - \mathsf{D}$ has been substituted by B and features U, V have been orthogonalized.

This way, the augmented Lagragian of (3.12) can be calculated much easier, as follows:

$$\begin{aligned} l(\mathsf{H},\mathsf{B},\mathsf{E},\mathsf{Z},\mathsf{N}) &= ||\mathsf{H}||_* + \alpha||\mathsf{B}||_* + \lambda||\mathsf{W}\circ\mathsf{E}||_1 \\ &+\langle \mathsf{Z}, \mathsf{X}-\mathsf{E}-\mathsf{U}\mathsf{H}\mathsf{V}^T\rangle + \frac{\mu}{2}||\mathsf{X}-\mathsf{E}-\mathsf{U}\mathsf{H}\mathsf{V}^T||^2{}_F \\ &+\langle \mathsf{N}, \mathsf{H}-\mathsf{B}-\mathsf{U}^T\mathsf{S}\mathsf{V}\rangle + \frac{\mu}{2}||\mathsf{H}-\mathsf{B}-\mathsf{U}^T\mathsf{S}\mathsf{V}||^2{}_F, \end{aligned} \tag{3.13}$$

where $||\mathsf{A}||_F$ is the Frobenius norm of a matrix A, $\mathsf{Z} \in \mathbb{R}^{n_1\times n_2}$ and $\mathsf{N} \in \mathbb{R}^{d_1\times d_2}$ are Lagrange multipliers and μ is the learning rate.

Subsequently, ADMM gets applied in (3.13) and generates a series of convex subproblems, which eventually produce the solution of the initial problem (3.11). All the subproblems are of the same format, meaning that at each ADMM iteration the variables $\mathsf{H}, \mathsf{B}, \mathsf{E}$ are updated serially, while the rest of them remain fixed. The unique solution of each

subproblem relies on the shrinkage $S_\tau(\mathsf{A})$ and the singular value thresholding $D_\tau(\mathsf{A})$ operators (with $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^T$ being the SVD of real matrix A), the usage of whom has already been analyzed during the solution of the convex problem (3.4). At the end of each ADMM cycle, Lagrange multipliers Z, N are updated as well.

More specifically, three convex subproblems need to be addressed to solve (3.13), creating a so called, 3-block separable convex objective. At first, minimizing (3.13) with regard to H at fixed $\mathsf{B}, \mathsf{E}, \mathsf{Z}, \mathsf{N}$ is identical to:

$$arg \min_{\mathsf{H}} l = arg \min_{\mathsf{H}} ||\mathsf{H}||_* + \mu||\mathsf{P} - \mathsf{UHV}^T||^2{}_F, \tag{3.14}$$

where $\mathsf{P} = \frac{1}{2}(\mathsf{X} - \mathsf{E} + \frac{1}{\mu}\mathsf{Z} + \mathsf{U}(\mathsf{B} + \mathsf{U}^T\mathsf{SV} - \frac{1}{\mu}\mathsf{N})\mathsf{V}^T)$. The solution of (3.14) is proved to be $\mathsf{U}^T\mathsf{D}_{\frac{1}{2\mu}}(\mathsf{P})\mathsf{V}$.

Respectively, minimizing (3.13) w.r.t. B is equivalent to:

$$arg \min_{\mathsf{B}} l = arg \min_{\mathsf{B}} \alpha||\mathsf{B}||_* + \frac{\mu}{2}||\mathsf{Q} - \mathsf{B}||^2{}_F, \tag{3.15}$$

where $\mathsf{Q} = \mathsf{H} - \mathsf{U}^T\mathsf{SV} + \frac{1}{\mu}\mathsf{N}$. Likewise, the update rule of (3.15) seems to be $D_{\frac{\alpha}{\mu}}(\mathsf{Q})$.

Furthermore, the same process is repeated w.r.t. E:

$$arg \min_{\mathsf{E}} l = arg \min_{\mathsf{E}} \lambda||\mathsf{W} \circ \mathsf{E}||_1 + \frac{\mu}{2}||\mathsf{R} - \mathsf{E}||^2{}_F, \tag{3.16}$$

where $\mathsf{R} = \mathsf{X} - \mathsf{UHV}^T + \frac{1}{\mu}\mathsf{Z}$. The kind of more complicated solution of (3.16) ends up being $S_{\lambda\mu^{-1}}(\mathsf{R}) \circ \mathsf{W} + \mathsf{R} \circ (\mathsf{1} - \mathsf{W})$.

Finally, the update of Lagrange multipliers occurs as follows:

$$\mathsf{Z} = \mathsf{Z} + \mu(\mathsf{X} - \mathsf{E} - \mathsf{UHV}^T) \tag{3.17}$$

$$\mathsf{N} = \mathsf{N} + \mu(\mathsf{H} - \mathsf{B} - \mathsf{U}^T\mathsf{SV}) \tag{3.18}$$

The whole mathematical reasoning we presented above, is summarised in Algorithm 2.

Algorithm 2 ADMM solver for PCPSFM

Input: Observation X, mask W, side information S, features $\mathsf{U},\ \mathsf{V}$, parameters $\alpha,\ \lambda > 0$, scaling ratio $\beta > 1$.

1: **Initialize:** $\mathsf{Z} = 0,\ \mathsf{N} = \mathsf{B} = \mathsf{H} = 0,\ \beta = \frac{1}{||\mathsf{X}||_2}$.
2: **while** not converged **do**
3: $\quad \mathsf{E} = S_{\lambda\mu^{-1}}(\mathsf{X} - \mathsf{UHV}^T + \frac{1}{\mu}\mathsf{Z}) \circ \mathsf{W} + (\mathsf{X} - \mathsf{UHV}^T + \frac{1}{\mu}\mathsf{Z}) \circ (\mathsf{1} - \mathsf{W})$
4: $\quad \mathsf{H} = \mathsf{U}^T D_{\frac{1}{2\mu}}(\frac{1}{2}(\mathsf{X} - \mathsf{E} + \frac{1}{\mu}\mathsf{Z}) + \mathsf{U}(\mathsf{B} + \mathsf{U}^T\mathsf{SV} - \frac{1}{\mu}\mathsf{N})\mathsf{V}^T))\mathsf{V}$
5: $\quad \mathsf{B} = D_{a\mu^{-1}}(\mathsf{H} - \mathsf{U}^T\mathsf{SV} + \frac{1}{\mu}\mathsf{N})$
6: $\quad \mathsf{Z} = \mathsf{Z} + \mu(\mathsf{X} - \mathsf{E} - \mathsf{UHV}^T)$
7: $\quad \mathsf{N} = \mathsf{N} + \mu(\mathsf{H} - \mathsf{B} - \mathsf{U}^T\mathsf{SV})$
8: $\quad \mu = \mu \times \beta$
9: **end while**

Return: $\mathsf{L} = \mathsf{UHV}^T, \mathsf{E}$.

It has become evident by now, that ADMM is able to solve complicated convex problems efficiently, within a small number of iterations. In this particular problem (3.11), a small number of iterations is of a great necessity, because of the high computation cost that occurs from certain steps of the algorithm. For example, the orthogonalization of the features U, V via the Gram-Schmidt process has an operation count of $O(n_1 d_1^2)$ and $O(n_2 d_2^2)$ respectively. At the same time, the update of matrix H in step 4 is the costliest computational operation of Algorithm 2. Specifically, the SVD required in the singular value thresholding action dominates with $O(min(n_1 n_2^2, n_1^2 n_2))$ complexity. To boost the performance even more, the authors have applied the fast continuation technique, which increases μ incrementally for accelerated superlinear performance [40]. The initialization strategies for variables H, B and Lagrange multipliers Z, N are described in [41]. Concerning the stopping criteria, the Karush-Kuhn-Tucker feasibility conditions have been employed, meaning if within a maximum number of 1000 iterations, the maximum of $||\mathsf{X} - \mathsf{E}_k - \mathsf{UH}_k\mathsf{V}^T||_F \,/\, ||\mathsf{X}||_F$ and $||\mathsf{H}_k - \mathsf{B}_k - \mathsf{U}^T\mathsf{SV}||_F \,/\, ||\mathsf{X}||_F$ dwindles from a predefined threshold ε, the algorithm is terminated. In this case, k signifies the index of the ongoing iteration.

Apart from the fast convergence, the 3-block separable convex objective of Algorithm 2 urges users to experiment with a number of different parameter combinations. For instance, if side information S is not available, PCPSFM reduces to PCP with features and missing values by setting α to zero. If the features U, V are not present either, PCP with missing values can be restored by fixing both of them at identity. Although, if only the side information S is accessible, without the features, the objective is transformed back into PCPSM. Later on, we will refer to three basic categories of parameterization we have used to conduct our experiments for the purposes of the face inpainting task.

4. EXPERIMENTAL EVALUATION AND DISCUSSION

4.1 Data Preparation

4.1.1 Dataset

For the purposes of our experiments, a part of the CelebFaces Attributes Dataset (CelebA) [27] was used, following the required processing to match the needs of the face inpainting task. CelebA is a large-scale face attributes dataset with 202,599 RGB celebrity face images, including 10,177 person identities. The images in this dataset cover large pose variations and background clutter. The reason behind the choice of CelebA as our test dataset is the fact, that LaFIn has already been trained on this exact dataset, so we had to make sure that PCPSFM would compete on equal terms. In fact, there are two versions of CelebA. The first, includes the initial, in-the-wild images as captured in real world conditions. For the needs of our project, we went for the second version, which comprises of the same images, but in a cropped and aligned format. This means, that only the face and sometimes a part of the upper body are depicted, while all images are aligned based on the five most important face landmark locations (left eye, right eye, nose, leftmost part of the mouth, rightmost part of the mouth). The alignment process is the outcome of face rotation, rescaling or transportation, as shown in some of the images of Figure 4.1. Occasionally it comes at the cost of ruining the background of the image.

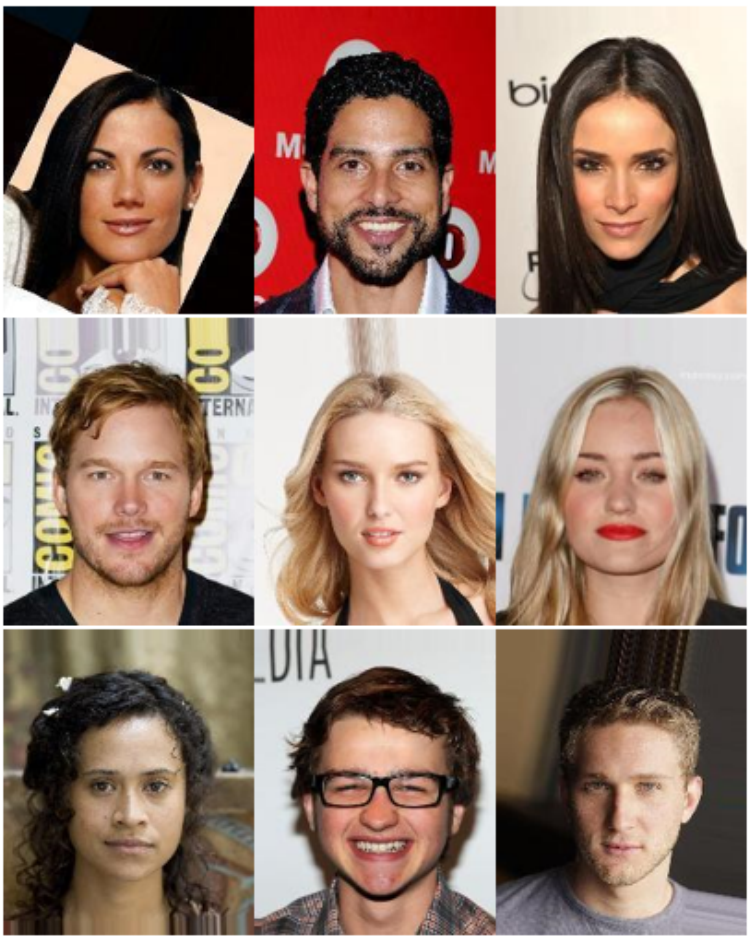

Figure 4.1: Sample of the aligned & cropped CelebA dataset

Even though, face alignment was a vital condition for the execution of our experiments, concerning mostly the PCPSFM method, we had to apply a more thorough processing, in order to isolate the faces completely and remove redundant information. This requirement occurred from the tendency of the face inpainting methods to confuse the background of image faces with extensive face occlusions, which usually leads to the generation of very unrealistic inpainted results. To overcome this obstacle, we proceeded to a manual cropping of the images, by cutting equal parts on the left side of the left eye and on the right side of the right eye, but also above the left eye and below the leftmost part of the mouth. By cropping equally an aligned image we ensured we would get a new aligned image of a smaller size. Specifically, the initial 218×178 pixels size images were transformed to 78×80 pixels size. Trying to accomplish as high resolution as possible we resized our dataset images to 100×100 pixels. After experimentation, this resolution was proven to be the highest possible we could use to perform the high memory demanding mathematical operations of the PCPSFM method, without dealing with memory crash issues. In Figure 4.2 there are depicted the manually cropped face images corresponding to the identities of Figure 4.1.

Figure 4.2: Sample of our dataset, including manually cropped CelebA face images

Concerning the number of images chosen from the CelebA dataset, it's clear we couldn't work with all of the 202,599 images, not only due to our restricted resources but also because of the long execution time required for each one of the multiple algorithm executions. Apart from, the quantity of the images, their quality is a crucial factor as well. For instance, we could not use en profile faces for the purposes of this project, where facial landmarks are not distinct, since the methods we used have to memorize as better as possible the basic structure and attributes of the human face, in order to effectively restore the occluded parts during the test procedure. Besides, the visual outcome of an en face image is far more captivating when the basic facial landmarks, like eyes, nose and mouth get successfully inpainted. For the reasons above, we picked out and processed 1600 images of 100 celebrity identities. The images are distributed equally, meaning there are

16 images per identity, including 1 manually occluded image. A sample of our dataset grouped per identity is shown in Figure 4.3.

Figure 4.3: Sample of our dataset grouped by identity

4.1.2 Occlusions

As mentioned above, our dataset contains one occluded face per identity. So, for the implementation of this project we applied 100 different occlusions, given that there are 100 identities located in our dataset. For the creation of the occlusions, we made use of the OpenCV python library. OpenCV is a library of Python bindings designed to solve computer vision problems. One of its uses is to design shapes of different sizes and colors on a given image. Thus, we tried to exploit OpenCV to create a variety of occlusion combinations of different shapes (rectangles, circles, triangles, lines) and sizes (small, medium, big), filled with realistic colors (shades of red, brown, pink) that could be detected in an actual face occlusion (e.g., a face trauma). For each one of the occluded images, our goal was to hide at least one of the significant facial landmarks, in order to make the inpainting task as challenging as possible for our models. In Figures 4.4 – 4.6, we present some occluded face images of our dataset, grouped by different occlusion criteria. The size criterion is the one we will use, later on, to evaluate the inpainting results of our methods.

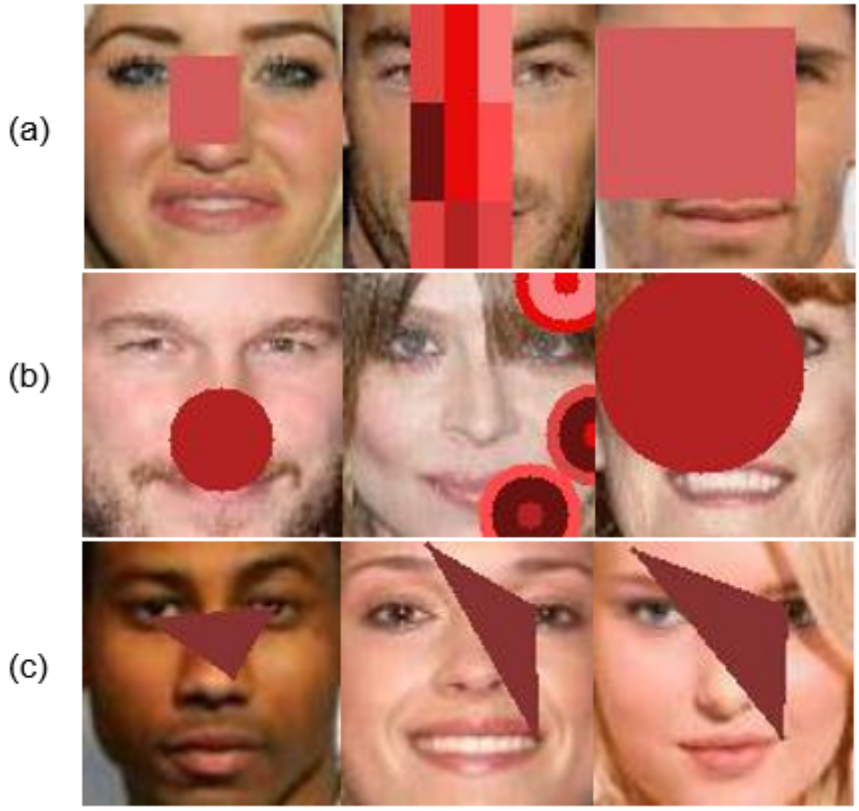

Figure 4.4: Occlusions grouped by shape. (a) Rectangular occlusions, (b) Circular occlusions, c) Triangular occlusions

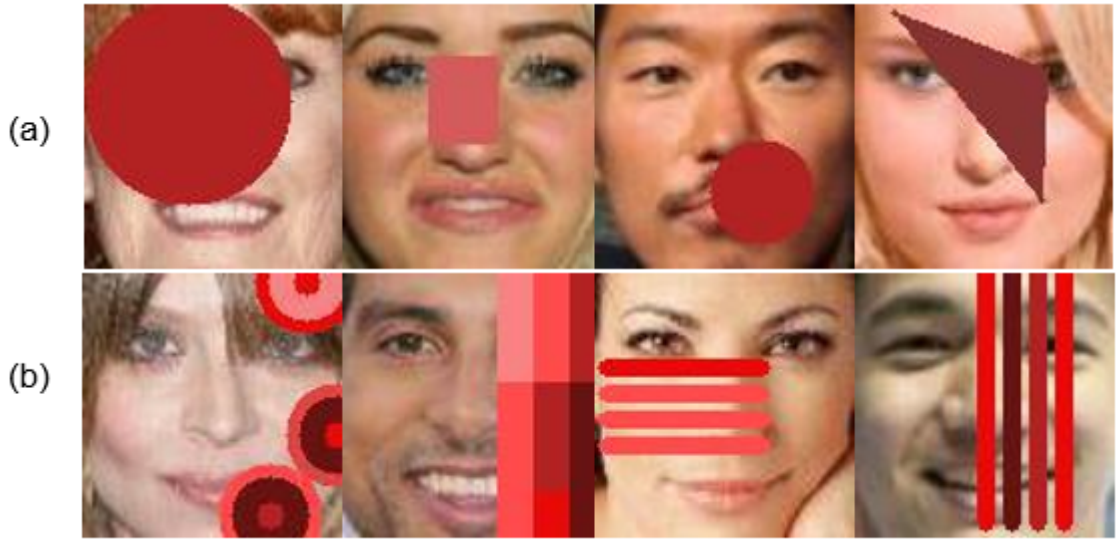

Figure 4.5: Occlusions grouped by sparsity. (a) Dense occlusions, (b) Sparse occlusions

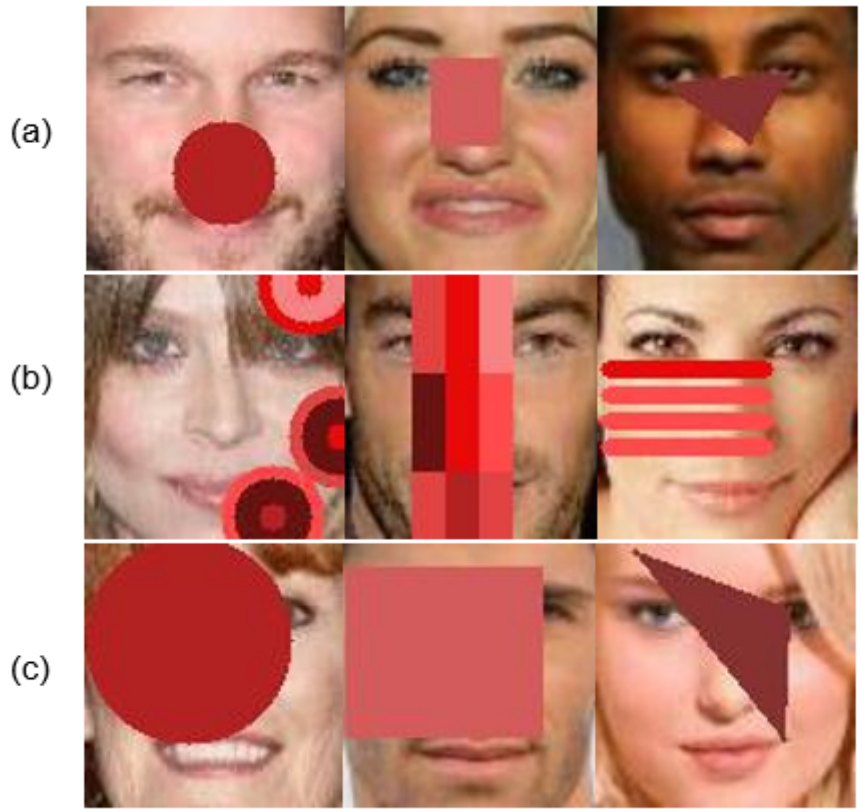

Figure 4.6: Occlusions grouped by size. (a) Small occlusions, (b) Medium occlusions, (c) Big occlusions

4.2 Models Setup

4.2.1 LaFIn

The implementation of LaFIn can be found in the github link of [17]. To experiment with LaFIn an NVIDIA GPU is required, to execute efficiently and within a short time, the complex machine learning models. In addition, a series of deep learning related Python libraries is needed, with the most significant being PyTorch, which specializes in the building of machine learning models from scratch. As known, GPUs are particularly expensive, so we had to conduct our experiments on the Google Collaboratory platform, which offers powerful GPUs for a limited amount of time per day.

Concerning LaFIn's configurations, the model can be set up for both training and test purposes, using either its own landmark predictor or FAN-Face landmarks. Moreover, it gives users the chance to decide, if they want to employ a mask or not, to cover the occluded part of the face for better inpainting results. Specifically, one of the following mask options may be employed:

- no mask
- random block mask
- center mask
- external mask
- 50% external, 50% random block mask
- 50% no mask, 25% random block, 25% external mask
- external non-random mask

For our face inpainting task we used a sole supervised, LaFIn model, initiated for test, using FAN-Face landmarks and external non-random masks. The use of a non-random mask is what makes the model supervised, because this way we have prior knowledge of where, precisely the occlusion is located and we can create an exact mask to cover it up. In Figure 4.7 there are depicted the masks that cover the occlusions of Figure 4.6.

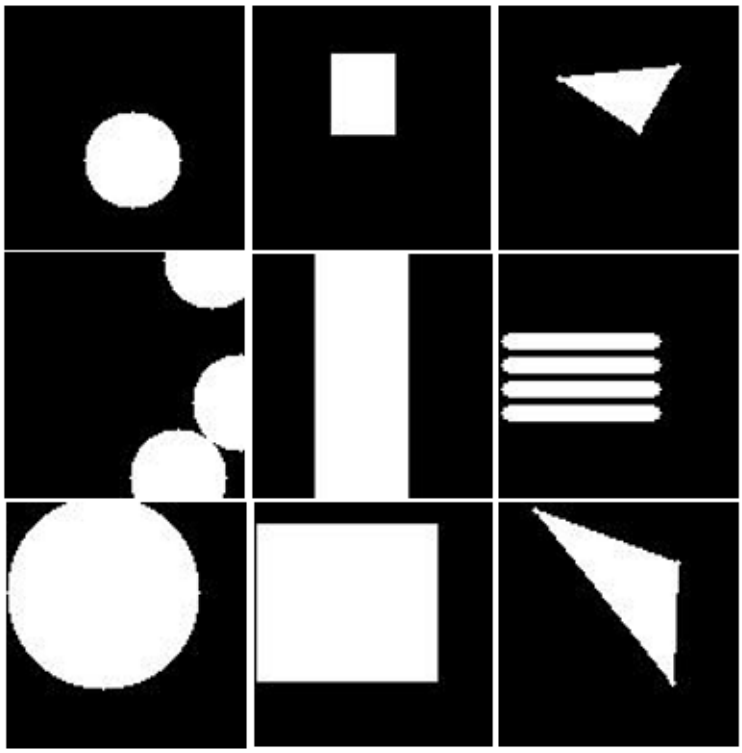

Figure 4.7: Sample of external non-random masks

4.2.2 PCPSFM

The implementation of PCPSFM was based completely on Algorithm 2, as presented in [21] and analyzed in subsection (3.4.2) of this thesis. For the implementation we used the Python programming language and the NumPy library, which specializes in complex mathematical operations between matrices. The only change we made to the algorithm is related to the value and the state of the scaling ratio factor β, that helps μ reach an accelerated superlinear performance. Particularly, after experimentation we noticed that the stable parameter $\beta = \frac{1}{||\mathsf{X}||_2}$, where X is the observation matrix, didn't lead to the best possible inpainting results. Hence, we decided to replace it, with a user-given value, which steadily decreases by 0.05 every 50 iterations, aiming to verge on $\beta = 1$, but without actually getting it done, so that the condition $\beta > 1$ is fulfilled. In other words, setting the value of β, as close to 1 as possible, is a crucial factor, which upgrades the quality of the inpainting results. However, small β values come at the cost of longer iterations, that may gradually lead to overextended execution times. For the reasons above, the values of β used during our experiments, satisfy the condition $1 < \beta \leq 1.5$.

Before getting into the remainder of parameter choices, we should first break down the three basic PCPSFM submethods we utilized in this project. At first, we tried to approach the inpainting problem via the perspective of the Classic RPCA (CRPCA) algorithm suggested in [35] and analyzed in section (3.3), meaning a purely unsupervised method without missing values but neither side information nor features. To achieve that, we had to create a huge observation matrix X, containing all the clear and occluded face images and at the same time we set the side information matrix S to zeros, the missing values matrix W to ones and the features matrices U, V to identity. The second, also, unsupervised method PCPF employs only the feature U and nothing else. In this case, feature U has the role of a database containing only clear face images, which indicate how non-occluded faces should look like. For this method to be functional, our initial dataset had to be split in half. In U there were located 8 clear face images per identity, meaning a total of 800 clear faces. The rest 800 clear and occluded face images were placed in X. S was set to zeros, W to ones and V to identity, as before. Our third PCPFM method is similar to the second one, but incorporates missing values too. The addition of missing values makes PCPFM semi-supervised, because similarly to LaFIn the information about the occlusion location is known, but the method can still remove occlusions, which haven't been pointed by the user. All matrices' values are identical to the PCPF ones, except from W, where we placed the missing values, representing the occlusion masks. The masks used in PCPFM are the same ones used to set up LaFIn (Figure 4.7).

Regarding the remainder of the parameters, the maximum number of iterations is set by default to 1000 for all the methods. Similarly, for all three cases the positive tuning parameter μ, used in augmented Lagrangian, the tolerance value for convergency ε and the positive tuning parameter α, used in the calculation of B have stable values. Specifically, $\mu = 10^{-5}$, $\varepsilon = 10^{-7}$ and $\alpha = 0$, since S is set to zeros in all cases. Furthermore, we initialized CRPCA with $\beta = 1.5$ and the other two methods with $\beta = 1.2$. We selected a higher β value for the CRPCA method, due to the fact that the observation matrix X is twice as big, compared to the respective X of PCPF and PCPFM, making the computational cost of Algorithm 2 significantly higher. Thus, by using a higher β value we were able to reduce the duration of each iteration, without causing a remarkable degradation to the inpainting results. The final PCPSFM models occurred after the initialization of each one of the three aforementioned submethods with three different λ inputs. Positive tuning parameter λ is used in the calculation of the sparse matrix E. Following the suggestion of [21] we

applied $\lambda = 1/\sqrt{max(n_1, n_2)}$, as the first λ value, where n_1, n_2 are the dimensions of the observation matrix X. In our case, $\lambda = 0.0057$, since $n_1 = 3000$ and $n_2 = 800\ or\ 1600$, depending on the executed method. Either way, n_1 is the maximum dimension and its value is the product of the image size (100×100) multiplied by 3, which represents the number of RGB channels. Dimension n_2 represents the number of images in X. Subsequently, we employed two more values $\lambda = 0.01$ and $\lambda = 0.1$, in order to have an overall view of the parameter's effect in our inpainting results. The configuration of all three PCPSFM methods can be summarized in Table 4.1.

Table 4.1: Initialization configuration of PCPSFM methods

Methods	Max iterations	α	μ	β	λ	ε
CRPCA	1000	0	10^{-5}	1.5	0.0057, 0.01, 0.1	10^{-7}
PCPF	1000	0	10^{-5}	1.2	0.0057, 0.01, 0.1	10^{-7}
PCPFM	1000	0	10^{-5}	1.2	0.0057, 0.01, 0.1	10^{-7}

4.3 Test Procedure

4.3.1 Models Execution

During the implementation of this thesis, all the experiments were executed in the Google Collaboratory platform. For the LaFIn related experiments the use of GPU was necessary, while for all the PCPSFM experiments the CPU sufficed. Therefore, as expected the execution of LaFIn was notably faster, given in fact that the network had to be fed only with the occluded images, since it was pre-trained on clear face images. On the other hand, PCPSFM models required all the face images for each unique execution, leading to longer execution times in all cases. Apart from the use of CPU, PCPSFM's rate was affected primarily by the size of the observation matrix X and secondarily by the values of the initialization parameters λ, β. Hence, a long range of different execution times was observed, not only among the three basic submethods, but also between a pair of models of the same method, depending exclusively on the unalike initialization values of their common parameters. Last but not least, we have to keep in mind that we evaluated the inpainting results based on the size of the occlusions. This means, we had to run all models for three different datasets, containing the same face images, but with different occlusions sizes (Figure 4.6). During our experiments, though, the occlusion size didn't seem to affect significantly neither the number of iterations nor the execution time of each model. As a result, in Table 4.2 we present the number of iterations and the average execution times of all our models, as occurred after the experimentation on each slightly modified dataset.

Table 4.2: Number of iterations & execution times of all models

Models		Iterations	Exec time (min)
LAFIN		-	3.5
CRPCA	$\lambda = 0.0057$	40	200
	$\lambda = 0.01$	45	240
	$\lambda = 0.1$	46	255
PCPF	$\lambda = 0.0057$	110	24
	$\lambda = 0.01$	119	25
	$\lambda = 0.1$	148	32
PCPFM	$\lambda = 0.0057$	110	19
	$\lambda = 0.01$	119	24
	$\lambda = 0.1$	148	25

4.3.2 Inpainting Results

In this section we display a sample of the results produced by our models for the face inpainting task and we comment on the models' efficiency, based on the visual outcomes of Figures 4.8 – 4.10. Our main purpose is to pursue the effect of the occlusion size in the face inpainting task. That's why, each figure illustrates the inpainting results of our models, as applied to the same, five face images for a different size of occlusion. The five celebrity faces selected, are on purpose of a totally different face structure and attributes, in order to help us acquire a wider view of the inpainting application in as realistic conditions as possible. Actually, identity V has the most heavily occluded face of our dataset, because of its preexistent occlusions (beard, eyeglasses), in addition to the manually created one. This raises an extra interest to our task, due to the fact that we didn't include the preexistent occlusions to the occlusion masks, which may lead to a completely different problem confrontation from the side of supervised, unsupervised and semi-supervised methods. Concerning the format of Figures 4.8 – 4.10, the first row illustrates the occluded face image and the second displays the ground truth image, meaning the pre-occluded image. In all the following rows, the inpainting results of all our models are depicted, grouped by the broad method, in which they belong (LAFIN, CRPCA, PCPF, PCPFM).

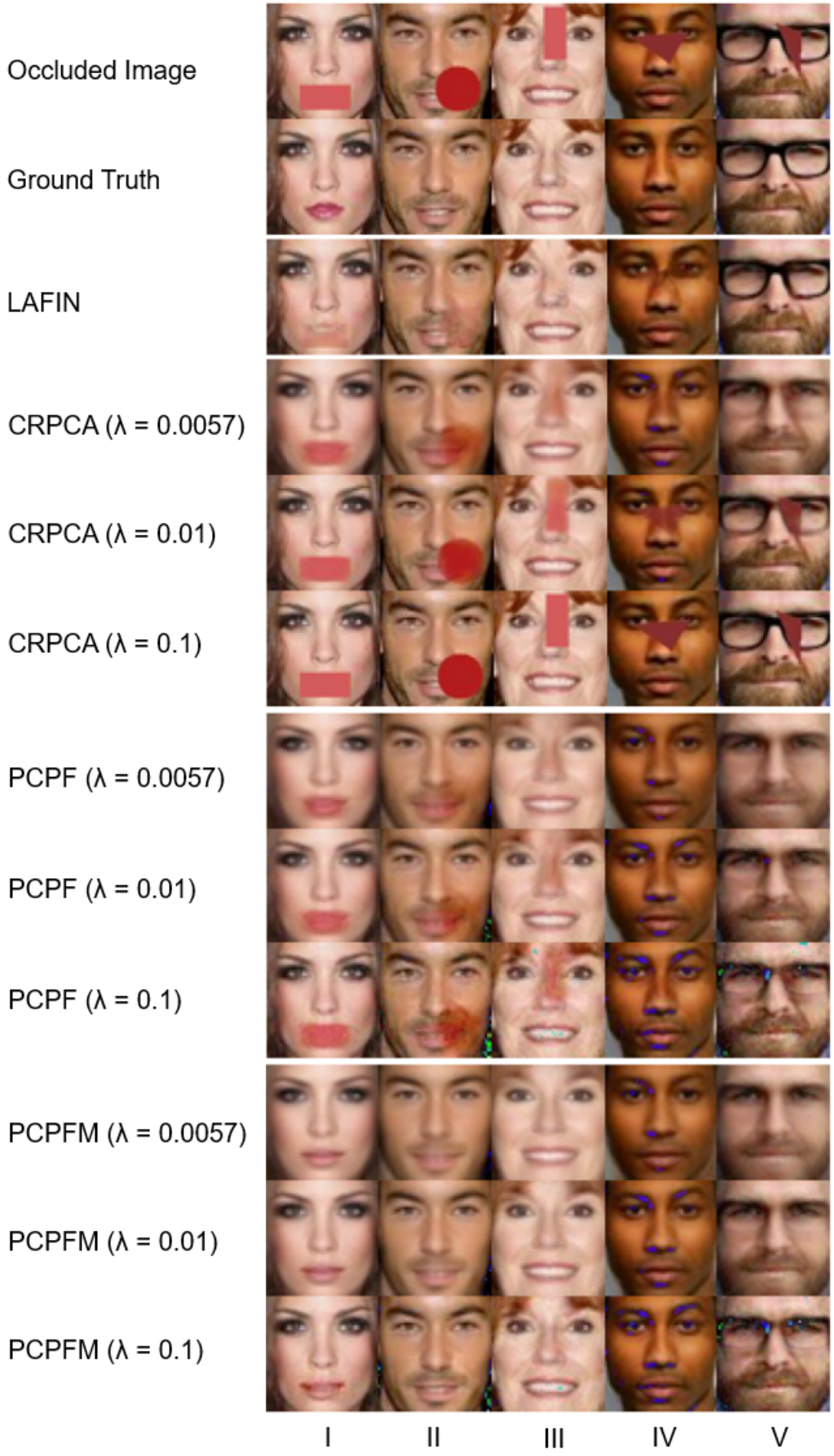

Figure 4.8: Image Inpainting on small occlusions

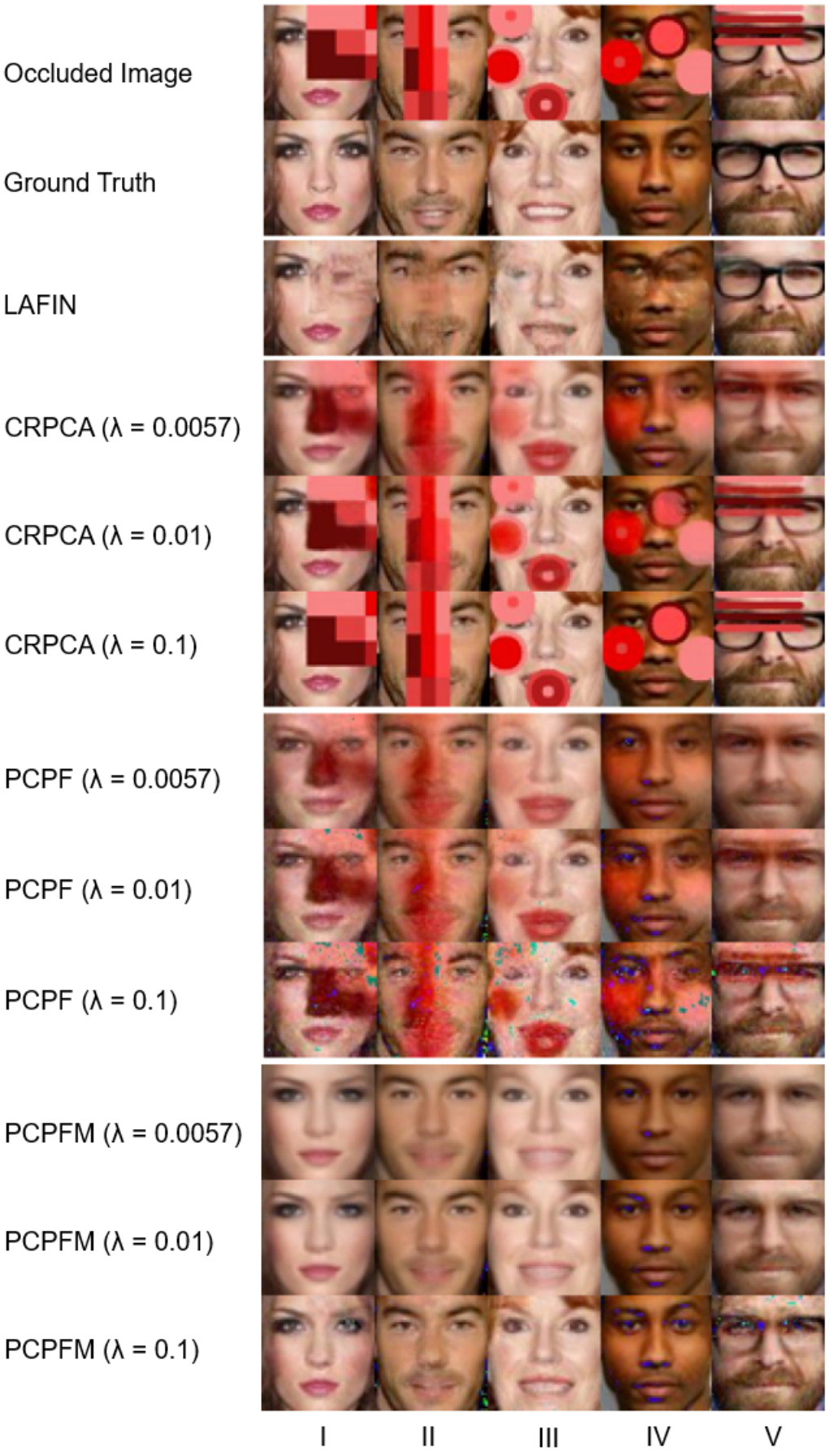

Figure 4.9: Image Inpainting on medium occlusions

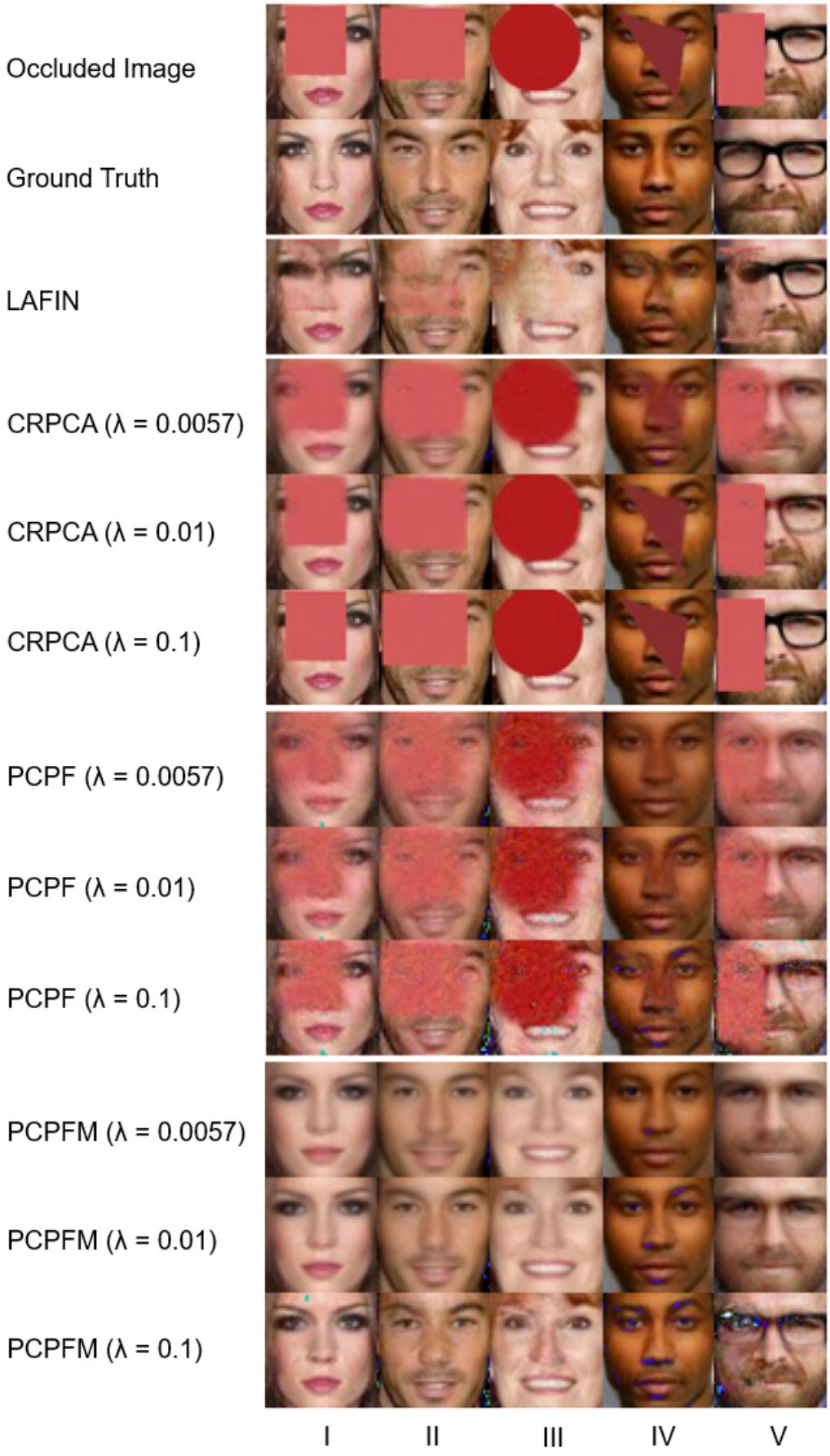

Figure 4.10: Image Inpainting on big occlusions

Numerous conclusions can be derived, after having a quick glimpse at the figures above. A first, general observation is that, as the size of the occlusions grows, the quality of the inpainting results downgrades. Especially in the case of small occlusions, all the models, except the CRPCA ones and PCPF ($\lambda = 0.1$), complete the inpainting process with high efficiency, being able to reproduce inpainted face images, almost identical to the ground truth. On the other hand, each one of the CRPCA models struggle to make the occlusion vanish, but none of them accomplishes it in an adequate grade. Actually, we could anticipate, that the results of CRPCA models would be deficient compared to the corresponding results of the other PCPSFM submethods. Besides, CRPCA's poor performance is the reason why side information and features had to be employed to upgrade the RPCA problem, at the first place. However, we should keep in mind that, due to its shy inpainting contribution, CRPCA has the advantage of maintaining some specialized face attributes intact, which will be proven to be a crucial factor in the upcoming evaluation results of the face recognition task.

At the same time, LAFIN sticks to a course very similar to CRPCA. Regarding, small occlusions LAFIN seems to produce equal, if not better inpainting results than the advanced PCPSFM models, like in the case of identity III. Though, as the occlusions grow bigger, it fails to reconstruct the covered face area, especially when this area contains one or more of the significant facial landmarks. Unlike CRPA models, which try to make the occlusion fade away without success, LAFIN shows a different kind of weakness. Particularly, it applies a type of blur in the occluded area, adopting the skin color of the examined identity. This process leads to a partially deformed face, which at least has a homogeneity, as regards the skin tone. We would probably expect better inpainting results from LAFIN, given the fact it has been trained on a huge number of images, containing all the identities we used in our dataset. However, we can still justify this mediocre performance, because as mentioned in subsection (4.1.1), the initial CelebA dataset includes a mixture of en face and en profile images, of a similar but non-identical structure and dimensionality, indicating a slightly different approach on the identification of the occlusions. Besides, the integrity of the clear facial parts combined with the skin homogeneity gain an upgraded role, when face recognition comes into play.

Meanwhile, the unsupervised method PCPF produces quite satisfying results for the case of small occlusions, as well as, for some individual cases of bigger occlusions, considering it has not prior knowledge of the occlusion location. The PCPF model initialized with $\lambda = 0.0057$ is the one that provides the best results of the method for all the occlusion categories. It's impressive, that PCPF can provide accurate results, even for just one out of five face images covered with big, sparse occlusions, just because the occlusion happens to be of a very similar color to the identity's skin (identity IV). Hence, we couldn't ask for a much better outcome from a method, which doesn't take into consideration the location of the occlusion, especially when the latter covers up to 40% of the face. However, PCPF's successful inpainting results don't come at zero cost. As analyzed in subsection (4.2.2), feature U has the role of a database full of clear faces, which provides PCPF the know-how about the structure and attributes of a non-occluded face. Then, PCPF utilizes this information as a prototype and reconstructs all the inpainted faces, based on it. In other words, as shown characteristically in the results of PCPF ($\lambda = 0.0057$) of Figure 4.8, the inpainted faces end up having almost identical face structure (in our case: eyes, eyebrows and nose), which of course doesn't keep up with reality. This face structure generalization will provoke a great confusion to the evaluation methods of the face recognition task, as we will see in the next section.

PCPFM is the final method, we experimented with and it's by far the best one, always with regard to the illustrations of Figures 4.8 – 4.10. We can't ascribe any significant misfire to PCPFM models, as they manage to vanish the occlusions and approach the ground truth, regardless of the examined occlusion size or identity. Although, the face structure generalization problem seems to still be present in some of the cases, as an aftereffect of feature U, PCPFM ($\lambda = 0.1$) model stands out and manages to restore the face attributes in remarkable detail. Therefore, the addition of missing values upgrades the preexistent PCPF method making it a powerful tool ready to confront the hardest of occlusions.

The last issue that needs to be addressed, is about the unique case of identity V. Previously, during the exploration of the models' inpainting results, we skipped on purpose to examine the ones related with this particular identity, because of its preexistent occlusions, which differentiate the expected outcome. Thus, we believe that identity V deserves a special mention, in order to clarify the course of action of each one of the supervised, unsupervised and semi-supervised methodologies, in the case of preexistent occlusions. Starting with the supervised, LAFIN model, the results don't surprise us. Particularly, concerning small and medium occlusions the inpainting process seems to work ideally and LAFIN generates results almost identical to the ground truth ones, with the preexistent occlusion remaining intact. Even though at the beginning of this project we didn't have the claim to deal with the case of preexistent occlusions, our unsupervised (PCPF) and semi-supervised (PCPFM) models took over the task for us. As a result, PCPF ($\lambda = 0.0057$) and PCPFM ($\lambda = 0.0057$) models generate almost the same, non-occluded faces, without eyeglasses and with a kind of trimmed beard, reminding us the clear faces of identity V, as depicted in Figure 4.3. In fact, this comparison concerns only the small occlusion dataset, because as mentioned before PCPF, doesn't perform well in larger types of occlusions. Namely, the small triangular occlusion can't prevent PCPF from generating high quality inpainting results equal to PCPFM. Of course, thanks to the introduction of missing values, PCPFM is able to deal with both preexistent and manual occlusions of all the shapes and sizes we experimented on.

To quantify and evaluate the quality of the inpainting results we applied a performance indicator in the form of a Reconstruction Error metric. This metric measures the distance between an inpainted image and the ground truth image and returns a value, which represents the deviation between the two images. It's clear, that the smaller the error value, the greater the similarity of the images. The reconstruction error is denoted as follows

$$RE = \left(\frac{||\mathsf{GT} - \mathsf{Inp}||_F}{||\mathsf{GT}||_F} \right)^2 \tag{4.1}$$

where GT is the array representation of the ground truth image and Inp is the array representation of an inpainted image. The array values lie in the range of [0,1].

For the evaluation of our inpainting results, we deployed the Mean Reconstruction Error (MRE), meaning we calculated the average reconstruction error for the 100 inpainted images produced by each model. In tables 4.3 – 4.5 we present the MRE values of our models for our three occlusion datasets.

Table 4.3: Mean Reconstruction Error on small occlusion dataset

Models		Mean Reconstruction Error
LAFIN		**0.007**
CRPCA	$\lambda = 0.0057$	0.023
	$\lambda = 0.01$	0.027
	$\lambda = 0.1$	0.033
PCPF	$\lambda = 0.0057$	0.025
	$\lambda = 0.01$	0.027
	$\lambda = 0.1$	0.038
PCPFM	$\lambda = 0.0057$	0.018
	$\lambda = 0.01$	0.015
	$\lambda = 0.1$	0.017

Table 4.4: Mean Reconstruction Error on medium occlusion dataset

Models		Mean Reconstruction Error
LAFIN		0.037
CRPCA	$\lambda = 0.0057$	0.076
	$\lambda = 0.01$	0.104
	$\lambda = 0.1$	0.127
PCPF	$\lambda = 0.0057$	0.066
	$\lambda = 0.01$	0.080
	$\lambda = 0.1$	0.115
PCPFM	$\lambda = 0.0057$	0.024
	$\lambda = 0.01$	**0.020**
	$\lambda = 0.1$	0.021

Table 4.5: Mean Reconstruction Error on big occlusion dataset

Models		Mean Reconstruction Error
LAFIN		0.039
CRPCA	$\lambda = 0.0057$	0.103
	$\lambda = 0.01$	0.105
	$\lambda = 0.1$	0.109
PCPF	$\lambda = 0.0057$	0.100
	$\lambda = 0.01$	0.102
	$\lambda = 0.1$	0.113
PCPFM	$\lambda = 0.0057$	0.026
	$\lambda = 0.01$	**0.022**
	$\lambda = 0.1$	0.026

4.4 Evaluation on Face Recognition

In the previous chapter we commented on the quality of the inpainting results produced by our models, based on the illustrations shown in Figures 4.8 – 4.10. However, modern applications, focusing on face images, don't flourish because of the possession of high-quality face images, but thanks to their ability to provide accurate predictions of the identities depicted in these images. Reaching at the end of this thesis, it has become clear by now, that the purpose of the face inpainting task is the restoration of occluded faces to a non-occluded form, in order to facilitate their identification. Though, we mustn't be complacent, that a visually flawless inpainted image guarantees the accurate retrieval of the identity.

Hence, for the purposes of this project we deployed three different evaluators to validate the accuracy of our inpainting models for the face recognition task. Before getting into each individual evaluator, we should first analyze the basic steps of the face recognition process. At first, we created a database of 800 face images, including 8 clear face images for each one of the 100 identities of our dataset. Then, we calculated and stored the face encoding of each of the 800 face images, creating a new database consisted of face encodings. A face encoding is an array of RGB values, containing specialized information obtained out of a face image. This information occurs from certain important measurements on the face, like the color, size and slant of eyes, the gap between eyebrows, the position of the nose and more. The same procedure had to be executed for the inpainted images, representing the test set of our experiments. For each inpainting model we calculated 100 different face encodings, extracted from the sole inpainted image of each identity. Afterwards, we passed the face encodings of the test set into an evaluator and for each one of them, the evaluator predicted one or more identity labels. Overall, each evaluator gets trained on the face encodings located in the database and tries to predict the identity shown in a given inpainted image by finding k database images with the most similar face encodings to the given image. Basically, the evaluator is a Classifier, who attempts to categorize the inpainted images, in 100 classes, namely the identities. To make it clear, we will give a simple example of a successful face classification. Suppose, we want to find the true identity behind identity II, by examining the inpainted image produced by LAFIN, as depicted in Figure 4.8. In other words, we want to classify one of the 100 inpainted images, as occurred after applying LAFIN inpainting on the small occlusion dataset. Let's also suppose, that we have initialized the Classifier to return only the most probable class label (k = 1). For the recognition to be accurate, the retrieved label must correspond to one out of 8 clear face images of the database, impersonating identity II. A schema of the inpainted image and the 8 clear face images of the database is depicted in Figure 4.11. The final step of the face identification process is the evaluation of the results. To achieve that we used two metrics. The first one is the Exact Accuracy metric, which checks if the most probable class label equals to the ground truth label. The second is the Ranked-K Accuracy metric, which examines if the ground truth label equals at least one of the top k most probable labels.

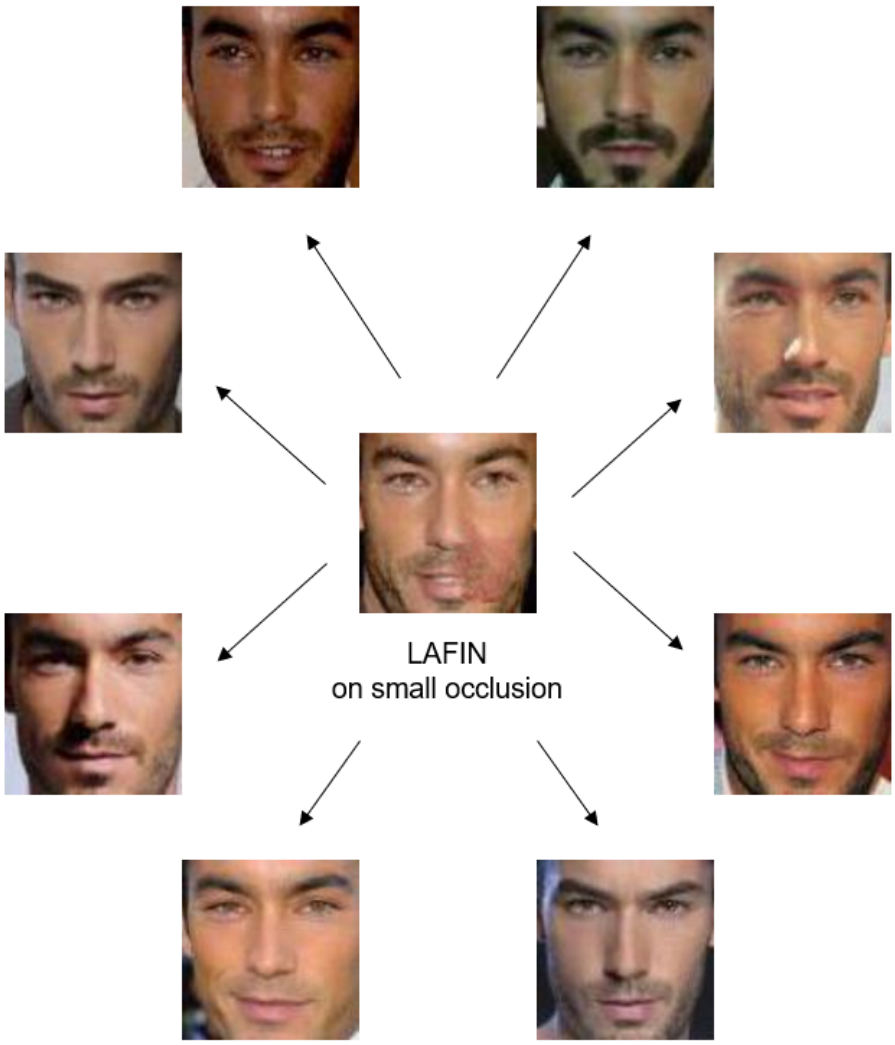

Figure 4.11: The 8 possible matches for an accurate prediction of the inpainted identity

4.4.1 K-Nearest Neighbors Classifier

The K-Nearest Neighbors (KNN) Classifier is based on the homonymous KNN algorithm. KNN is a simple, supervised machine learning algorithm that can be used to solve classification problems. It assumes that similar things exist in close proximity, namely near to each other. At first, KNN stores all the initial data (train set) and then classifies the new data points (test set), based on their similarity to the initial data. This means, when a new data point appears it can be easily classified into the most similar of the available categories. During our experiments, we used the KNeighborsClassifier implemented in the Scikit-learn Python package. For the generation of face encodings, we utilized the face_recognition Python library, which constructs an array of 128 values, representing the unique face attributes. In Tables 4.6 – 4.8 we present the KNN classification results for our three occlusion size cases, as calculated using the exact accuracy and raked-K accuracy metrics, where k = 3, 5 stands for the number of nearest neighbors of the KNN algorithm.

Table 4.6: Evaluation results on small occlusions using KNN Classifier

Models		Exact accuracy (%)	Ranked-3 accuracy (%)	Ranked-5 accuracy (%)
LAFIN		**92**	**96**	**97**
CRPCA	$\lambda = 0.0057$	81	89	90
	$\lambda = 0.01$	85	94	95
	$\lambda = 0.1$	87	92	94
PCPF	$\lambda = 0.0057$	57	74	79
	$\lambda = 0.01$	72	78	83
	$\lambda = 0.1$	65	75	77
PCPFM	$\lambda = 0.0057$	62	76	78
	$\lambda = 0.01$	82	87	89
	$\lambda = 0.1$	80	85	86

Table 4.7: Evaluation results on medium occlusions using KNN Classifier

Models		Exact accuracy (%)	Ranked-3 accuracy (%)	Ranked-5 accuracy (%)
LAFIN		66	71	80
CRPCA	$\lambda = 0.0057$	58	69	73
	$\lambda = 0.01$	68	72	77
	$\lambda = 0.1$	62	73	80
PCPF	$\lambda = 0.0057$	42	48	53
	$\lambda = 0.01$	45	55	60
	$\lambda = 0.1$	33	43	49
PCPFM	$\lambda = 0.0057$	48	62	67
	$\lambda = 0.01$	69	77	84
	$\lambda = 0.1$	**76**	**83**	**85**

Table 4.8: Evaluation results on big occlusions using KNN Classifier

Models		Exact accuracy (%)	Ranked-3 accuracy (%)	Ranked-5 accuracy (%)
LAFIN		51	60	63
CRPCA	$\lambda = 0.0057$	31	40	49
	$\lambda = 0.01$	44	54	60
	$\lambda = 0.1$	44	55	60
PCPF	$\lambda = 0.0057$	18	26	33
	$\lambda = 0.01$	23	34	37
	$\lambda = 0.1$	26	34	36
PCPFM	$\lambda = 0.0057$	45	55	58
	$\lambda = 0.01$	59	**71**	**77**
	$\lambda = 0.1$	**64**	70	70

4.4.2 Linear SVM Classifier

Linear Support Vector Machine (or Linear SVM) is another supervised machine learning model suitable for the solution of classification problems. Its main attribute is the creation of a line or a hyperplane, which separates the data into classes. In the SVM algorithm, the initial data (train set) are mapped as points in a n-dimensional space with their values being the coordinates of their locations. The objective of SVM is to maximize the width of the gap between the two classes. The new data points (test set) are then mapped into that same space and they join the class, which corresponds to the side of the hyperplane they fall into. For our experiments, we used the SVC Classifier, initialized with a linear kernel and implemented in the Scikit-learn Python package, as well. Once again, the face encodings were generated using the face_recognition Python library. Tables 4.9 – 4.11 illustrate the Linear SVM classification results for our three occlusion size cases, as calculated using the exact accuracy metric. Due to the nature of the SVM Classifier we cannot apply the ranked-K accuracy metric.

Table 4.9: Evaluation results on small occlusions using Linear SVM Classifier

Models		Exact accuracy (%)
LAFIN		**96**
CRPCA	$\lambda = 0.0057$	79
	$\lambda = 0.01$	87
	$\lambda = 0.1$	90
PCPF	$\lambda = 0.0057$	62
	$\lambda = 0.01$	74
	$\lambda = 0.1$	69
PCPFM	$\lambda = 0.0057$	67
	$\lambda = 0.01$	79
	$\lambda = 0.1$	82

Table 4.10: Evaluation results on medium occlusions using Linear SVM Classifier

Models		Exact accuracy (%)
LAFIN		60
CRPCA	$\lambda = 0.0057$	33
	$\lambda = 0.01$	68
	$\lambda = 0.1$	67
PCPF	$\lambda = 0.0057$	40
	$\lambda = 0.01$	44
	$\lambda = 0.1$	32
PCPFM	$\lambda = 0.0057$	61
	$\lambda = 0.01$	69
	$\lambda = 0.1$	**76**

Table 4.11: Evaluation results on big occlusions using Linear SVM Classifier

Models		Exact accuracy (%)
LAFIN		54
CRPCA	$\lambda = 0.0057$	33
	$\lambda = 0.01$	45
	$\lambda = 0.1$	47
PCPF	$\lambda = 0.0057$	23
	$\lambda = 0.01$	22
	$\lambda = 0.1$	24
PCPFM	$\lambda = 0.0057$	42
	$\lambda = 0.01$	**66**
	$\lambda = 0.1$	65

4.4.3 VGGFace2 Classifier

VGGFace2 Classifier is an implementation of our own. Practically, we built a KNN Classifier, named after the VGGFace2 dataset. VGGFace2 is made of around 3.31 million images divided into 9131 classes, each representing a different person identity. Thanks to its low label noise, high pose and age diversity, it has become a popular dataset suitable to train state-of-the-art deep learning models on face-related tasks. We named our Classifier after VGGFace2, because this time, we chose to generate the face encodings in a different way. Specifically, we deployed a ResNet50 [46] neural network pre-trained on VGGFace2, which takes an image face as input and returns an array of 2048 values, representing the unique face attributes, namely the face encodings. Afterwards, we passed the face encodings in our KNN Classifier, where we utilized the cosine similarity metric to classify our inpainted results to the predicted identities. Similarly to our first KNN Classifier, in Tables 4.12 – 4.14 we present the classification results of our VGGFace2 Classifier for all the occlusion cases, calculated using the exact accuracy and raked-K accuracy metrics.

Table 4.12: Evaluation results on small occlusions using VGGFace2 Classifier

Models		Exact accuracy (%)	Ranked-3 accuracy (%)	Ranked-5 accuracy (%)
LAFIN		**88**	**95**	**96**
CRPCA	$\lambda = 0.0057$	78	86	87
	$\lambda = 0.01$	84	91	93
	$\lambda = 0.1$	86	92	93
PCPF	$\lambda = 0.0057$	61	69	74
	$\lambda = 0.01$	71	82	84
	$\lambda = 0.1$	73	80	88
PCPFM	$\lambda = 0.0057$	64	75	83
	$\lambda = 0.01$	81	88	90
	$\lambda = 0.1$	82	92	**96**

Table 4.13: Evaluation results on medium occlusions using VGGFace2 Classifier

Models		Exact accuracy (%)	Ranked-3 accuracy (%)	Ranked-5 accuracy (%)
LAFIN		75	82	85
CRPCA	$\lambda = 0.0057$	54	65	72
	$\lambda = 0.01$	71	81	85
	$\lambda = 0.1$	75	82	87
PCPF	$\lambda = 0.0057$	37	54	61
	$\lambda = 0.01$	46	56	65
	$\lambda = 0.1$	43	55	61
PCPFM	$\lambda = 0.0057$	46	62	70
	$\lambda = 0.01$	67	79	84
	$\lambda = 0.1$	**76**	**84**	**88**

Table 4.14: Evaluation results on big occlusions using VGGFace2 Classifier

Models		Exact accuracy (%)	Ranked-3 accuracy (%)	Ranked-5 accuracy (%)
LAFIN		**61**	**74**	**78**
CRPCA	$\lambda = 0.0057$	30	42	48
	$\lambda = 0.01$	43	58	65
	$\lambda = 0.1$	46	65	76
PCPF	$\lambda = 0.0057$	19	28	33
	$\lambda = 0.01$	27	35	39
	$\lambda = 0.1$	32	38	42
PCPFM	$\lambda = 0.0057$	38	53	61
	$\lambda = 0.01$	51	66	75
	$\lambda = 0.1$	59	73	**78**

4.4.4 Interpretation of Classification results

Observing the accuracy results of our three evaluators we realize that in all cases, except one, the classifiers point out the same inpainting model as the most dominant. However, we can't distinct a sole classifier as the superior for all three occlusion sizes. In fact, the divergence between the classification percentages of the corresponding models is negligible among the classifiers. Starting from the case of small occlusions, we notice a clear supremacy of the LAFIN model. Particularly, Linear SVM Classifier achieves an exact accuracy score of 96% for the LAFIN inpainting results, which is the highest accomplished among all evaluation results. Actually, Figure 4.8 had predisposed us for LAFIN's superiority and PCPFM' strong performance. Though, after a 1 – 1 examination of the models, based on the λ parameter, we realize that PCPFM models are inferior to the CRPCA ones. At first sight, this is an unexpected outcome, judging from the results depicted in Figure 4.8. Yet, as we mentioned before, CRPCA's weak inpainting contribution, combined with the small occlusion size amplifies the preservation of the most significant face attributes, extracted in the form of face encodings. This seems to gradually lead to a more accurate face recognition. At last, the face structure generalization issue prevents PCPF from reaching the standards set by the other methods, leading to rather low evaluation results, even for the small occlusion dataset.

Continuing with the examination of the evaluation results, we perceive that, as the occlusions grow bigger PCPFM begins to stand out, which sounds logical, considering it is the only method able to remove the occlusions from every single face image. Specifically, as analyzed before, the PCPFM models and particularly the one initialized with $\lambda = 0.1$ seem to overcome the face structure generalization problem. For that reason, the latter ends up producing the most accurate results for the medium occlusion dataset, setting the bar up to 76% for the exact accuracy metric. The fact that all classifiers accomplish the same highest accuracy score for the LAFIN model, shows their diversity and reliability in evaluating complex machine learning tasks, such as face recognition. So, if we had to pick only one out of three classifiers for this specific occlusion case, we would probably choose VGGFace2 Classifier, as it is the one which achieves a slightly better classification (84%) for the respective PCPFM model, based on the second most significant metric, namely the ranked-3 accuracy. Generally, the ranked-K accuracy metric, is a very useful metric for the purposes of the face recognition task, as it unfolds the efficiency of each model in a broad manner, even if it doesn't always refer directly to the ground truth classification class. Coming back to our results, VGGFace2 Classifier seems to be the most stable classifier in this particular occlusion case, as it achieves exact accuracy rates close to 75% for three models built on three different methods. This is not the case, for the other two classifiers, each of which achieves just one exact accuracy score over 70%. Last but not least, despite the redistributions in the accuracy hierarchy the majority of CRPCA models keep maintaining the second place, this time above LAFIN, while the PCPF ones still hold the final positions, with the percentages experiencing a deep fall.

Finally, regarding the case of big occlusions a controversy occurs from the classification results, about which inpainting model is the most appropriate for the purposes of face recognition. Specifically, KNN and Linear SVM Classifiers nominate the PCPFM method as the superior, which is an expected outcome, having examined the illustrations in Figure 4.10. On the other hand, VGGFace2 Classifier, surprisingly points out LAFIN as the most suitable model, leaving PCPFM as the second-best option. As always, we justify our choices using the exact accuracy metric as guidance. Having said that, we go for the suggestions of the first two classifiers, given the fact that they both achieve scores near 65%, for their best PCPFM model, while VGGFace2 Classifier slightly surpasses 60% for the LAFIN model. Particularly, Linear SVM achieves 66% for the PCPFM method, initialized with $\lambda = 0.01$ this time. However, due to the large extension of the occlusions in this dataset, most exact accuracy scores are notably below 50% for CRPCA and PCPF methods. Therefore, we can verify with numbers and reinforce the general observation we made in the previous section. Namely, as the size of the occlusions grows, not only the quality of the inpainting results downgrades, but also the efficiency in solving the face recognition task.

5. CONCLUSION AND FUTURE WORK

The purpose of this thesis was to restore a number of occluded face images to a non-occluded form, using different inpainting methods, which we evaluated on a face recognition task. LaFIn is a supervised method built upon a series of deep neural networks, which integrate facial landmarks and generate an inpainting result for a given occluded face image. During our experiments, we instantiated LaFIn with the FAN-Face model, suitable for the prediction of landmarks and we created our own external non-random masks to cover the occluded parts of each face. PCPSFM, on the other hand, is an unsupervised method, based on the RPCA methodology. It is fed with both occluded and non-occluded face images and returns a low-rank matrix L_0, containing all the inpainted face images and a sparse matrix S_0, which consists of images, depicting the occluded parts of the initial faces. To achieve that, PCPSFM incorporates a number of improvements, compared to the classic RPCA method, with the most important of them being side information, features and missing values. In this project, we implemented three PCPSFM variations, two totally unsupervised and one semi-supervised, aiming to experiment with different combinations of the introduced improvements. We initialized all three submethods with an adequate number of parameters. Some of them were identical among the methods and others were unique for each model, to make it stand out from the rest. In the end, a total of ten models occurred from the two primary methodologies. During test, we deployed these models on three manually created datasets, containing the same face images, but with an increasing occlusion size, dataset after dataset. The test procedure resulted in a discussion about the effectiveness of each model, in regard to the quality of its inpainting results. This discussion involved some unexpected inpainting outcomes, that helped us subtle the difference between the supervised, unsupervised and semi-supervised methodologies. Finally, we utilized all the generated inpainted images to simulate a face recognition system, similar to the ones used in modern face-related applications. To evaluate the quality of our results, we employed three classifiers (KNN Classifier, Linear SVM Classifier, VGGFace2 Classifier), each built upon a unique methodology. Through the evaluation process, we acquired a spherical perception of each model's efficiency and we apprehended how the occlusion size affects each method's functionality. Finally, we concluded that there isn't a universal, superior inpainting method, suitable for all the occlusion cases. Hence, we pointed out the most appropriate model for each individual case, based on the results provided by our classifiers.

To sum up, through the course of this thesis we managed to produce a number of successful inpainting results and we achieved very satisfying evaluation scores on the face recognition task, always in proportion to the examined occlusion size. However, there are still many upgrades we could have included in our experiments to optimize even more the inpainting results and the evaluation accuracy, if we had the appropriate resources and a loose deadline. First of all, having access to a stable machine with an integrated GPU, would allow us to focus on a more complex implementation of Algorithm 2, designed to utilize the processing power of the GPU, which would lead to a notable reduction of PCPSFM's execution time. In this context, it would be in our best interest to process and experiment with a lot more than 1600 face images, in order to feed our models with as many data as possible, expecting a significant improvement in the inpainting results. In fact, the face images could be of a resolution higher than 100×100 to expose in even more detail the restoration of the occluded face parts. Moreover, taking for granted the availability of resources in the future, it may be worth proceeding to the re-training of LaFIn network, on the exact same cropped and aligned set of clear face images, used as side

information or features in the PCPSFM models. This way, the comparison of the two methods will be fair, counter to the circumstances of this project, where we made use of the default LaFIn version, pre-trained on the initial in-the-wild, aligned CelebA images. Furthermore, crucial conclusions can be extracted, about the models' functionality, if we group our dataset by additional occlusion criteria, apart from the size. For instance, as analyzed in subsection (4.1.2) we could sort the occlusions, depending on their shape, sparsity or even the location they lie on the face image. Hence, by combining all these criteria we will get a deeper perception of how the inpainting methods respond to each type of occlusion and it will be easier for us to detect the most demanding occlusion cases. Last but not least, there is always a chance of finding a better evaluation method, capable of achieving higher accuracy scores for the face recognition task.

ABBREVIATIONS - ACRONYMS

ADMM	Alternating Direction Method of Multipliers
ALM	Augmented Lagrange Multiplier
CelebA	CelebFaces Attributes Dataset
CRPCA	Classic Robust Principal Component Analysis
FAN	Face Alignment Network
FRN	Face Recognition Network
FW-PCA	Fast Weighted Principal Component Analysis
GAN	Generative Adversarial Network
KNN	K-Nearest Neighbor
LaFIn	(Generative) Landmark Guided Face Inpainting
LSTM	Long Short-Term Memory
MRE	Mean Reconstruction Error
PCA	Principal Component Analysis
PCP	Principal Component Pursuit
PCPF	Principal Component Pursuit with Features
PCPFM	Principal Component Pursuit with Features and Missing Values
PCPSM	Principal Component Pursuit with Side Information and Missing Values
PCPSFM	Principal Component Pursuit with Side Information, Features and Missing Values
RNN	Recurrent Neural Network
RPCA	Robust Principal Component Analysis
ReLU	Rectified Linear Activation Unit
RGB	Red, Green, Blue
SISR	Single Image Super-Resolution
SVD	Singular Value Decomposition
SVM	Support Vector Machine

REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. “Image inpainting”. In 27th Annual Conference on Computer Graphics and Interactive Techniques, pages 417–424, 2000.

[2] H. Yamauchi, J. Haber, and H.-P. Seidel. “Image restoration using multiresolution texture synthesis and image inpainting”. In Computer Graphics International, pages 120–125. IEEE, 2003.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: “A randomized correspondence algorithm for structural image editing”. ACM Transactions on Graphics (ToG), 28(3):24:1–24:11, 2009.

[4] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf. “Image completion using planar structure guidance”. ACM Transactions on graphics (TOG), 33(4):129:1–129:10, 2014.

[5] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. “Context encoders: Feature learning by inpainting”. In CVPR, pages 2536–2544, 2016.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. “Generative adversarial nets”. In NeurIPS, pages 2672–2680, 2014.

[7] S. Iizuka, E. Simo-Serra, and H. Ishikawa. “Globally and locally consistent image completion”. ACM Transactions on Graphics (ToG), 36(4):107, 2017.

[8] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. “Generative image inpainting with contextual attention”. In CVPR, pages 5505–5514, 2018.

[9] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. “Free-form image inpainting with gated convolution”. arXiv preprint arXiv:1806.03589, 2018.

[10] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. “Image inpainting for irregular holes using partial convolutions”. In ECCV, pages 85–100, 2018.

[11] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi. “Edgeconnect: Generative image inpainting with adversarial edge learning”. arXiv preprint arXiv:1901.00212, 2019.

[12] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo. “Foreground-aware image inpainting”. In CVPR, pages 5840–5848, 2019.

[13] Y. Li, S. Liu, J. Yang, and M.-H. Yang. “Generative face completion”. In CVPR, pages 3911–3919, 2017.

[14] Y. Jo and J. Park. “Sc-fegan: Face editing generative adversarial network with user’s sketch and color”. arXiv preprint arXiv:1902.06838, 2019.

[15] J. Zhang, X. Zeng, Y. Pan, Y. Liu, Y. Ding, and C. Fan. “Faceswapnet: Landmark guided many-to-many face reenactment”. arXiv preprint arXiv:1905.11805, 2019.

[16] Q. Sun, L. Ma, S. Joon Oh, L. Van Gool, B. Schiele, and M. Fritz. “Natural and effective obfuscation by head inpainting”. In CVPR, pages 5050–5059, 2018.

[17] Y. Yang, X. Guo, J. Ma, L. Ma, H. Ling. “LaFIn: Generative Landmark Guided Face Inpainting”. arXiv:1911.11394. github: https://github.com/YaN9-Y/lafin

[18] Electronic Frontier Foundation, July 1990. https://www.eff.org/pages/face-recognition [Accessed 2/9/21]

[19] A. Criminisi, P. Perez, and K. Toyama. “Region filling and object removal by exemplar-based image inpainting”. IEEE Transactions on image processing, 13(9):1200–1212, 2004.

[20] L. Cheng, J. Wang, Y. Gong, and Q. Hou. “Robust deep auto-encoder for occluded face recognition”. In Proceedings of the 23rd ACM international conference on Multimedia, pages 1099–1102. ACM, 2015.

[21] N. Xue, J. Deng, S. Cheng, Y. Panagakis, S. Zafeiriou. “Side Information for Face Completion: A Robust PCA Approach”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(10):2349 – 2364, 2019.

[22] A. Kumar and R. Chellappa. “Disentangling 3d pose in a dendritic cnn for unconstrained 2d face alignment”. In CVPR, pages 430–439, 2018.

[23] S. Xiao, J. Feng, L. Liu, X. Nie, W. Wang, S. Yang and A. Kassim. “Recurrent 3d-2d dual learning for large-pose facial landmark detection”. In CVPR, pages 1633–1642, 2017.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In CVPR, pages 4510–4520, 2018.

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenk, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In CVPR, 2017.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”. In CVPR, 2016.

[27] Z. Liu, P. Luo X. Wang, X. Tang, “Large-scale CelebFaces Attributes (CelebA) Dataset”, 29 July 2016; https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html [Accessed 6/9/2021]

[28] O. Ronneberger, P. Fischer, T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. arXiv: 1505.04597.

[29] C. Zheng, T.-J. Cham, and J. Cai. “Pluralistic image completion”. In CVPR, pages 1438–1447, 2019.

[30] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. “Image to-image translation with conditional adversarial networks”. In CVPR, pages 1125–1134, 2017.

[31] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. “Spectral normalization for generative adversarial networks”. arXiv preprint arXiv:1802.05957, 2018.

[32] S. Hochreiter, J. Schmidhuber, 1997. “Long short-term memory”. Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[33] I. Jollie. “Principal Component Analysis”. Springer-Verlag, 1986.

[34] D.P. Berrar, W. Dubitzky, M. Granzow. “Singular value decomposition and principal component analysis”. In A Practical Approach to Microarray Data Analysis, pages 91-109, 2003. LANL LA-UR-02-4001.

[35] E. J. Candes, X. Li, Y. Ma, J. Wright. “Robust Principal Component Analysis?” arXiv:0912.3599.

[36] X. Yuan and J. Yang. “Sparse and low-rank matrix decomposition via alternating direction methods”. Preprint, 2009.

[37] D.P. Bertsekas. “Constrained Optimization and Lagrange Multiplier Method”. Academic Press, 1982.

[38] K. Chiang, C. Hsieh, and I. Dhillon, “Robust principal component analysis with side information”. In ICML, 2016.

[39] F. Shang, Y. Liu, J. Cheng, and H. Cheng, “Robust principal component analysis with missing data”. In CIKM, 2014, pages 1149–1158.

[40] R. T. Rockafellar, “Monotone operators and the proximal point algorithm”. SIAM journal on control and optimization, vol. 14, no. 5, pages 877–898, 1976.

[41] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers”. Foundations and Trends in Machine Learning, vol. 3, no. 1, pages 1–122, 2011.

[42] J. Yang, A. Bulat, G. Tzimiropoulos. “FAN-Face: a Simple Orthogonal Improvement to Deep Face Recognition”. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 12621-12628. doi: 10.1609/aaai.v34i07.6953.

[43] A. Bulat and G. Tzimiropoulos, G. 2017a. “Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources”. In ICCV.

[44] A. Bulat and G. Tzimiropoulos, G. 2017b. “How far are we from solving the 2d and 3d face alignment problem?”. In ICCV.

[45] A. Newell, K. Yang and J. Deng. 2016. “Stacked hourglass networks for human pose estimation”. In ECCV.

[46] K. He, X. Zhang, S. Ren, and J. Sun. 2016. “Deep residual learning for image recognition”. In CVPR.