University of Nevada, Reno

Doctoral Dissertation

Mitigating Environmental Bias in Facial Recognition Models

Author:
Nathan Thom

Dissertation Advisor:
Emily Hand

A dissertation submitted in partial fulfillment of the requirements for the degree of Ph.D. in Computer Science and Engineering

Computer Vision and Machine Perception
Department of Computer Science and Engineering

April 2024

THE GRADUATE SCHOOL

We recommend that the dissertation
prepared under our supervision by

Jay Thom

entitled

AI Enabled IoT Network Traffic Fingerprinting with Locality Sensitive Hashing

be accepted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

Advisor
Shamik Sengupta, Ph.D.

Committee Member
Frederick Harris, Ph.D.

Committee Member
Batyr Charyyev, Ph.D.

Committee Member
Emily Hand, Ph.D.

Committee Member
Hanif Livani, Ph.D.

Graduate School Representative

Markus Kemmelmeier, Ph.D., Dean
Graduate School

Abstract

Abstract....

Acknowledgement

Acknowledgement....

Contents

List of Figures

Chapter 1

Introduction

1.1 Thesis Organization

1. Introduction
2. Background
3. Related Work
4. Parsing Faces with Semantic Segmentation
5. Consensus Subspace Clustering
6. Deep Vision Model Perception of Gender From Faces
7. DoppelVer: A Benchmark for Face Verification
8. Studying Representations of Facial Recognition Models in Visually Similar Environments
   OR
9. Data Driven Attributes
10. Conclusion

11. Future Research

1.1.1 Introduction

We present a doctoral dissertation on the improvement of facial attribute recognition with deep convolutional neural networks. We explore the recognition of traditional facial attributes and propose a method for improving their recognition. Additionally, we propose methods for automatically assigning labels to image data, explore the perceptions of CNNs, expand the pool of available facial attributes to novel data driven attributes, and finally present a challenging benchmark for facial verification which we believe will allow for improvement to facial recognition methods.

In Chapter 4 we present a method of improving traditional attribute recognition. This is achieved by supervising the recognition task with weak semantic segmentation labels. The effectiveness of this method alludes to the importance of facial geometry to the task of attribute recognition. Enforcing that the CNN recognize the correct location of the attribute along with the presence of the attribute is valuable. This also suggests that other methods of attribute recognition could be identifying spurious correlations amongst visual features and the target attribute. For example, identifying other parts of face which commonly co-occur with the target attribute. This is on a surface level not too egregious, but the downstream uses of facial attributes will likely be affected. We also identify evidence that improvement of traditional facial attribute recognition is a data problem as much as it is a methodological issue.

In Chapter 5 we present an effort to both increase the volume of attribute data which is available and reduce the noise which exists in the attribute labels. We propose a method of unsupervised psedo labeling called Consensus Subspace Clustering (CSC). CSC filters image features with a group of dimensionality reduction

techniques. This filtering results in a set of features which are used to group samples into meaningful classes without prior knowledge of the images. We utilize spectral clustering with consensus to provide higher quality pseudo labels. While effective for some types of visual data, this approach demonstrates limitations in ability to effectively model fine-grained visual data. Because of this we pursue a deeper understanding of the perceptions and representations of deep vision models.

In Chapter 6 we present of a study of the perceptions of deep vision models when performing the task of gender recognition. In addition, we present a method for explaining the predictions of vision models performing face-related tasks. This work reveals the regions of most value to CNNs performing face-related tasks and provides insight into their performance when facial regions are occluded. The task of improving the data problem of facial attributes remains.

In Chapter **??**, we present a method of producing valuable facial attributes from a small pool of images labeled with identity. We refer to this method as Data Driven Attributes (DDAs). DDAs are extracted by training individual classifiers to perform the binary classification task of differentiating between two identities. We generate many such classifiers to produce a set of attributes which is valuable in the same scenarios as traditional attributes. Example use cases of attributes are facial recognition, image search, etc. While these attributes are valuable in many ways we believe that their descriminative power would be improved by the formation of pairs with high visual similarity. That is, face images which are different in particular ways. Such pairs are found in doppelganger pairs.

In Chapter 7 we present a novel dataset of visually similar image pairs called DoppelVer. The dataset contains two protocols of similar pairs. The first protocol - Doppelganger - contains positive image pairs which are images depicting the same identity, but negative image pairs which are images depicting a doppelganger identity. The second protocol - ViSE - features negative image pairs which are

identities who would not be mistaken as each other, but in this one-off case appear as highly similar. We demonstrate that state-of-the-art face recognition models struggle to accurately perform face verification on both protocols.

Chapter 2

Background

Background....

Chapter 3

Related Works

Related work.....

Chapter 4

Parsing Faces with Semantic Segmentation for Improved Facial Attribute Recognition

4.1 Introduction

Facial attribute recognition was first introduced as a means to improve face verification performance [4–6]. Face verification is the problem of identifying whether or not two images contain the same person. Now a field in its own right, facial attribute recognition focuses on classifying human describable features of faces in images or video. Prior to deep learning, research in facial attribute recognition included the FaceTracer face search engine [4], simile classifiers on the PubFig dataset [5], and multi-label classification on the Labeled Faces in the Wild (LFW) dataset [7].

In 2015, deep learning methods became popular for facial attribute recognition, with the introduction of two large-scale datasets for the problem: CelebA and LFWA [1]. With the introduction of these large scale datasets, the number of

deep learning methods for the problem of attribute recognition exploded. Although CelebA allowed for significant progress to be made in the field, it has been shown to have significant label imbalance, with many of the methods based on this dataset relying on correlations between unbalanced features [8**?**]. To address this issue, we propose a joint learning architecture in which attribute recognition is combined with semantic segmentation – a task that is independent of inter-class correlations.

Semantic segmentation is the problem of classifying every pixel in an image as belonging to one or multiple classes. State of the art methods for semantic segmentation utilize convolutional neural networks (CNNs) and seek to identify a single class for each pixel in an image [9–11]. Very few works address the problem of semantic segmentation of faces. Kalayeh et al. propose segmenting the face into parts for improved attribute recognition [12]. This however still differs from our approach, which segments faces according to where each attribute occurs on the face. Thus, we have generated a novel, weak labeling of attribute segments for the CelebA dataset. This enables learning of attribute localization alongside attribute recognition.

The proposed work introduces a novel technique for facial attribute recognition. A basic facial attribute recognition model is strengthened with additional supervision from a weakly labeled semantic segmentation task. Segmentation labels are generated essentially for free by automatically extracting facial landmarks, then a rule-based system uses landmarks to label the portions of input images where attributes occur. Prior work has shown that many state-of-the-art algorithms rely very heavily on attribute correlations, rather than the actual presence of an attribute [8**?**]. We address this problem through the combination of weakly supervised semantic segmentation and attribute recognition in one learning framework. By generating weak segmentation labels for each attribute, our method learns where to look for an attribute as well as what to look for when recognizing attributes of a face. We

show that this multi-task framework leads to an improved representation of facial attributes which does not merely rely on correlations between classes.

To summarize, this work's contributions include:

- AttParseNet: a multi-task CNN for simultaneous attribute localization and recognition using a weakly labeled training approach.
- A framework for generating semantic segmentation labels in the context of facial attributes
- Weak attribute segments for the full CelebA dataset, to be released with the publication of this work.

4.2 Related Work

The proposed research combines work in semantic segmentation and attribute recognition. We detail the relevant literature in the following sections.

4.2.1 Semantic Segmentation

Semantic segmentation is a fundamental task in computer vision that involves assigning a label to each pixel in an image. This technique has found widespread applications in various domains, such as autonomous driving [13, 14], pedestrian detection [15, 16], and computer-aided diagnosis [17, 18]. Although some traditional methods of segmantic segmentation exist [19, 20], the field is dominated by deep learning and CNNs. Pioneering works, such as Fully Convolutional Networks (FCNs) [21] and U-Net [22], have demonstrated remarkable performance and have set the foundation for subsequent research in this field.

Face parsing, also known as facial semantic segmentation, is a specialized sub-field of semantic segmentation that focuses on segmenting facial regions into semantically meaningful parts, such as skin, hair, eyes, nose, and mouth. Traditionally, Conditional Random Fields were used by all state-of-the-art methods for face parsing [23–25]. As in many other fields, deep learning became the new state-of-the-art [26–30].

4.2.2 Attribute Recognition

Facial attribute recognition was first introduced by Kumar et al. in [4]. The same group later showed that attributes were useful for search and retrieval as well as face verification [5, 6]. Early works in this domain focused on handcrafted features, such as SIFT [31] and HOG [32], coupled with traditional machine learning classifiers, including Support Vector Machines (SVM) and AdaBoost.

In 2015, Liu et al. introduced the large-scale benchmark dataset CelebA, containing over 200,000 images each labeled with 40 binary attributes [1].The introduction of CelebA was a significant milestone for the field because its size enabled the use of deep learning methods. Sample images from CelebA are shown in Figure 4.4. Along with CelebA, Liu et al. introduced a method for face localization and attribute recognition that involves two networks: LNet and ANet [1]. LNet performs localization for faces with weak attribute supervision, and ANet uses the localized face to predict facial attributes.

[8**? ?**] recently introduced methods to combat label imbalance for the problem of attribute recognition. The Mixed Objective Optimization Network (MOON) from [8] addressed label imbalance by calculating source and target distributions for each attribute and applying a weight to the backpropagation to adjust for the difference between the distributions. As a follow-up to MOON, [**?**] introduced a method called "Selective Learning" where balancing is performed at the batch

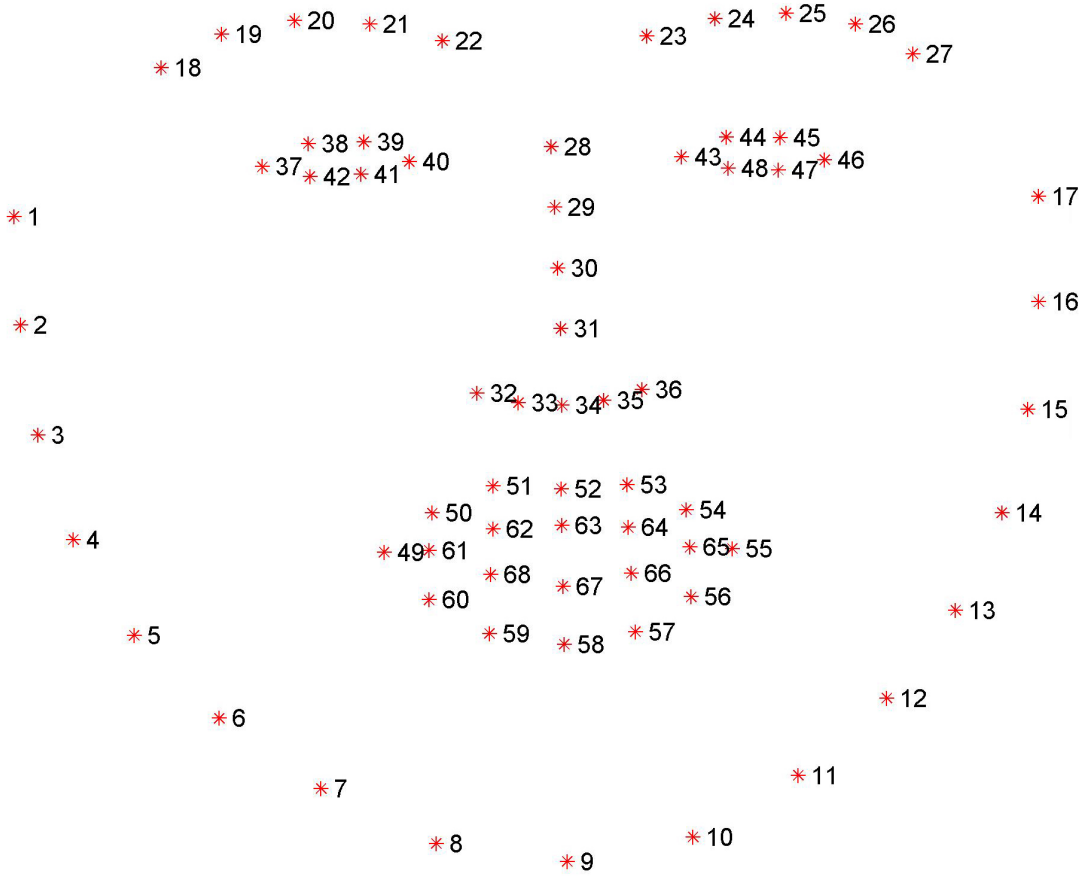


FIGURE 4.1: Layout of facial landmarks extracted from OpenCV and OpenFace.

level by weighing underrepresented classes and sampling from over-represented ones.

This work utilizes the task facial parsing as a weak form of weak supervision for the task of facial attribute classification. To be clear, facial parsing is learned alongside facial attribute classification to enforce that the network learn to locate the coarse spatial position of each attribute. Few works that we are aware of utilize facial parsing in this way. An example of one such work focuses on segmenting parts using hand-labeled data for training [12]. Kalayeh et al. implement novel pooling and gating mechanisms that utilize face parsing labels generated by a separate network. Our work differs significantly in that we use a multi-task learning framework to perform attribute segmentation and recognition in the same network.

4.3 Proposed Methods

The proposed method consists of two main parts: the generation of weak segmentation labels, and the multi-task learning framework. In the following sections we detail each.

4.3.1 Segmentation Label Generation

Teaching the model to localize facial attributes is facilitated by semantic segmentation labels. This form of labeling assigns classes to each pixel in an input image. For the scope of this work, the pixels of each image in CelebA are labeled with the presence or absence of 40 attribute classes. The binary labels for the classes are provided along with the CelebA dataset. Example attribute classes are *smiling*, *wavy hair*, *young*, etc. Our semantic segmentation labels are represented as masks of the same height and width as the input images and a depth of 40 channels (one for each attribute class). Segment masks have a value of 255 in regions where the attribute is present and 0 everywhere else. Hand-labeling this data is expensive and slow, so we opt to automate the process by introducing a weak labeling strategy, which requires no human supervision.

Generation of segment labels begins by extracting a set of facial landmarks from each image in CelebA. Figure 4.1 shows the layout of the 68 facial landmarks that are used. We utilize the OpenCV and OpenFace landmark detectors to extract these points [33, 34].

OpenCV's facial landmark detector is a pre-trained model that localizes 68 key facial points in an image. It extracts Histogram of Oriented Gradients (HOG) features and applies a cascade of regression trees to iteratively refine the landmark positions, starting from an initial estimate. OpenFace's landmark detector is a pre-trained set of CNNs which produce response maps without knowledge of other

landmark positions. These response maps are produced from expert models at a variety of scales and and angles. These response maps are considered jointly to produce point estimates for each landmark.

In both cases, the final output is a set of 68 facial landmark coordinates that accurately identify the jawline, eyebrows, nose, eyes, and mouth. This approach is fast, efficient, and robust to variations in facial pose, expression, and lighting conditions. We first pass all images through the OpenCV detector. Images which do not receive predictions are passed through the OpenFace detector. This technique yields fiducial points for over 99% of CelebA. The remaining images are hand-labeled with landmarks.

The set of collected 68 facial landmarks are used to define a set of base facial regions. The base regions are **below chin**, **chin**, **cheeks**, **mouth**, **above mouth**, **nose**, **eyes**, **eyebrows**, **ears**, and **top of head**. The **chin**, **mouth**, **nose**, **eyes** and **eyebrows** regions are precise because they are defined directly from the 68 landmark points. The remaining 5 regions are established by combining these precise regions with information about facial geometry. For example, the **top of head** region is created by using landmarks from the eyebrows and information about facial geometry, since no landmarks for the forehead are given. We refer to these regions as *rough segments*. Figure 4.2 shows the different regions used in the generation of attribute segments. For example, the **mouth** region is defined as the polygon which has vertices at landmark points {49-60}.

Each of the 40 attribute labels in CelebA will be mapped to a set of base regions which contain the visual features necessary for detecting a given attribute. We assume that the attribute *Smiling* occurs in the base region of **mouth**. Some attributes, such as *No_Beard*, are located in multiple facial regions: **below chin**, **chin**, **cheeks**, and **above mouth**.

FIGURE 4.2: Examples of the 10 base regions used to generate weak semantic segmentation labels. These regions are overlayed with the original images for visualization purposes. The segment regions are show in blue and landmark points are red.

Combining this information with the attribute labels in CelebA enables a nearly automatic system for producing segmentation labels for the entire dataset. This method is significant because it provides a framework for producing additional layers of supervision on arbitrary attribute recognition tasks.

To generate the segmentation masks we begin by referencing the 40 attribute labels provided with the CelebA dataset. Each segmentation mask begins as a black image, all pixels set to 0. If the attribute is labeled as present, we fetch the base regions which are the attribute is mapped to. The correct polygons are formed and filled with pixel values of 255. Each image in CelebA receives 40 segmentation masks, resulting in over 8 million segmentation masks total.

We consider the segmentation labels to be weak for two reasons: 1) our rule-based method for generating segments relies on automated facial landmark extraction, which may result in imprecise landmarks and regions, and 2) the physical manifestation of several attributes is unclear, leading to proposed segments that may not provide adequate coverage. To clarify the two types of weak labels in our

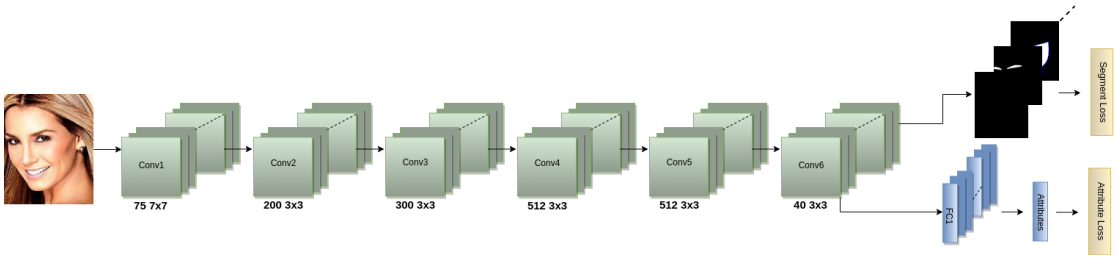


FIGURE 4.3: Our multi-task learning architecture. Input of an image is provided and is passed through 6 shared convolutional layers. The network outputs segmentation masks and attribute predictions.

segmentation work, we provide several examples. For type-1 segments, misaligned mouth landmarks can lead to incorrect mouth segments. Moreover, the absence of hair landmarks means that all hair-related attributes (e.g., *brown hair*, *wavy hair*) have rough segments derived from the **top of head** region. For type-2 segments, there is ongoing debate in the field of expression and micro-expression recognition regarding the indicators of a smile: whether it is solely the mouth or if other facial deformations around the eyes also play a role. In this work, we assume that the mouth is responsible for mouth-related attributes, potentially missing out on other facial cues.

4.3.2 Attribute Segmentation and Recognition

Once the weakly labeled attribute segments have been generated, the next step is to build a model that learns to recognize attributes. Attribute recognition and segmentation are learned jointly with a CNN architecture that we call AttParseNet. The task of semantic segmentation is used to improve our model's attribute recognition accuracy and generalizability.

The proposed multitask attribute segmentation and recognition model is an eight-layer CNN. The architecture for the CNN is shown in Figure 4.3. The model consists of six convolution layers, the first using filters of size 7x7, and the remaining layers using filters of size 3x3. The number of filters in each layer is as

follows: 75, 200, 300, 512, 512, and 40. Max pooling is performed after the first convolution layer. After the final convolution layer, the model produces 40 feature maps, each of size 96x76. Each feature map represents a facial attribute and the location in which it occurs.

The generated segmentation labels are the same size as the input images. To compare the masks with the feature maps from the final convolution layer, we perform downsampling to a size of 96x76. The downsampling operation utilizes nearest neighbor interpolation, which assigns each pixel in the downsampled image the value of the nearest pixel in the original image without any averaging or blending. This method is chosen to retain the binary nature of the segmentation masks, as it preserves sharp edges and avoids introducing intermediate values. These downsampled feature maps are then passed into two loss computation modules: the semantic segmentation loss and the facial attribute recognition loss.

The semantic segmentation loss is formulated as mean squared error (MSE) between the output feature maps and segment labels. In this context, MSE loss is used to measure the reconstruction error between the predicted segmentation masks and the ground truth masks. Given an input image, the model generates C feature maps of size $h\times$ w, $where C represents the number of attributes or classes in the dataset. Thes$

$$\text{MSE} = \frac{1}{C}$$

$\times$ h $\times$ w $\sum_{c=1}^{C} \sum_{i=1}^{h} \sum_{j=1}^{w} (y_{c,i,j} - \hat{y}_{c,i,j})^2$

where $y_{c,i,j}$ represents the value of the downsampled ground truth mask at position (i, j) for attribute c, and $\hat{y}_{c,i,j}$ represents the corresponding predicted value from the model's output feature map. By minimizing the MSE loss during training, the model learns to generate segmentation masks that closely match the ground truth masks, thereby improving its ability to identify the spatial location of visual features necessary for attribute recognition.

For the recognition task, the feature maps are flattened but not concatenated. Each flattened feature map is passed into a separate fully connected layer with a shape of 7296x1. This results in a final 40-dimensional output prediction. The facial attribute recognition loss is calculated using binary cross-entropy (BCE) loss. BCE loss is commonly used for multi-label classification tasks, where each sample can belong to multiple classes simultaneously. In this case, each facial attribute is treated as an independent binary classification problem. The model predicts the presence or absence of each attribute based on the flattened feature maps.

Given a batch of N images, the BCE loss for the facial attribute recognition task is computed as follows:

$$\text{BCE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}[y_{n,c}\log(\hat{y}_{n,c}) + (1 - y_{n,c})\log(1 - \hat{y}_{n,c})]$$

where $y_{n,c}$ is the ground truth label for attribute c in image n, and $\hat{y}_{n,c}$ is the predicted probability of attribute c being present in image n. The BCE loss penalizes the model for incorrect predictions and encourages it to learn the correct attribute labels.

For AttParseNet to learn from both the segmentation and recognition tasks in one framework, each task has its own loss function. During training, the total loss for the model is computed as an equally weighted sum of the semantic segmentation loss and the facial attribute recognition loss: Total Loss = MSE + BCE

The proposed multitask CNN architecture offers several advantages. By sharing features across both tasks, the model can learn more robust and generalized representations. The weak semantic segmentation task in AttParseNet provides an added level of supervision to the problem of attribute recognition for free. By "free," we mean that there is a very small amount of human labeling required,

and the segments are generated using facial landmark points and weakly labeled using the image-level attribute labels provided with CelebA.

Adding weakly labeled semantic segmentation to AttParseNet forces the model to activate on regions of interest when learning attribute representations, which leads to a more robust and generalizable attribute model. We showcase this in our experiments. **It is important to note that the weak segments are used only at training time and are not needed during testing.**

4.4 Experiments and Results

4.4.1 Datasets

We begin our experimentation on the CelebA dataset, a large-scale face attributes dataset commonly used for facial attribute recognition tasks. Introduced by Liu et al. [1], CelebA consists of 202,599 celebrity face images, each annotated with 40 binary attributes such as gender, age, hair color, and facial features like smiling or wearing glasses. The dataset was carefully curated to provide a diverse set of images with varying poses, backgrounds, and lighting conditions, making it a challenging and representative dataset for evaluating the performance of attribute recognition models. Example images can be seen in Figure 4.4

The images in CelebA were sourced from the internet and cover a wide range of real-world scenarios. The dataset is divided into three subsets: a training set containing approximately 162,000 images, a validation set with 20,000 images, and a test set with the remaining 20,000 images. This split allows for proper model development, hyperparameter tuning, and unbiased evaluation of the final trained models.

FIGURE 4.4: Sample images from the CelebA dataset [1].

One notable aspect of the CelebA dataset is that it features both cropped and aligned images, as well as full body, unaligned images. In our experiments, we crop the full body, unaligned images with out extracted landmark points. These images are used for training AttParseNet, as this allows the model to learn from more natural and unconstrained facial representations. For training the baseline network, we use the cropped and aligned images, which provide a more focused view of the facial region. Both AttParseNet and the baseline network require input images with dimensions of 218x178 pixels. Therefore, we resize the cropped images and segmentation labels to 218x176 and 96x76 pixels, respectively.

To further validate the generalization capability of our trained models, we also evaluate their performance on two additional datasets: LFWA [1] and UMD-AED [2]. The LFWA dataset contains 13,232 face images of 5,749 identities, annotated with the same 40 attributes as in CelebA. We report results on the entire LFWA dataset to assess how well our models perform on a different distribution of images. The UMD-AED dataset, despite its modest size of 2,800 facial images,

FIGURE 4.5: Sample images from the LFWA dataset [1].

proves to be a powerful tool for exposing vulnerabilities in attribute models. Each image is annotated with a subset of the 40 attributes found in CelebA and LFWA datasets. A unique characteristic of UMD-AED is its balanced distribution of positive and negative samples for each attribute, with 50 instances of each. This equilibrium enables the dataset to effectively uncover the limitations of attribute models. Images of LFWA and UMD-AED can be found in Figures 4.5 and 4.6 respectively.

By evaluating our models on multiple datasets with varying characteristics, we aim to provide a comprehensive analysis of their robustness and ability to generalize to different domains.

4.4.2 AttParseNet Training

AttParseNet is trained exclusively on the unaligned, cropped images from the CelebA training split, without any additional external data. This approach ensures that the model learns to extract relevant features and perform attribute recognition solely based on the information available within the CelebA dataset.

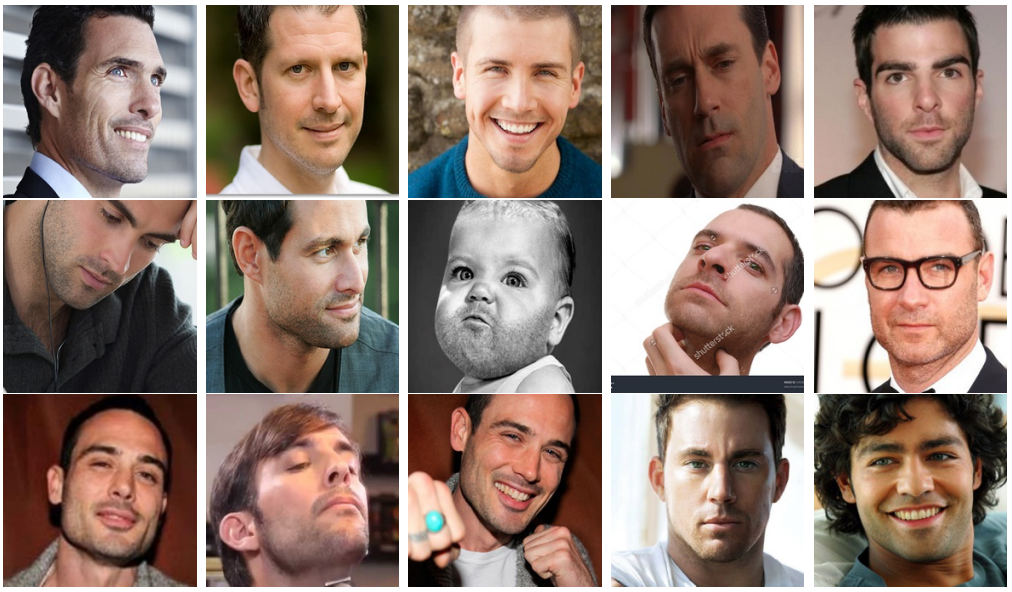


FIGURE 4.6: Sample images from the UMD-AED dataset [2].

The training process for AttParseNet consists of two stages. In both stages, we use the Adam optimizer [35] with a learning rate of 1E-3 to update the network weights. The Adam optimizer is chosen for its adaptive learning rate capabilities and efficient convergence properties.

In the first stage, the model is trained for 10 epochs, during which the network weight updates are based solely on the MSE loss computed from the semantic segmentation task. The semantic segmentation task involves predicting a coarse segmentation mask for each facial attribute, providing a high-level understanding of the spatial distribution of attributes. By training the model initially on this task, we allow the network weights to warm up and converge to a reasonable starting point. This stage helps in reducing the MSE loss to a level where the values of the BCE loss become comparable, facilitating effective joint learning in the subsequent stage.

The second stage of training involves a multi-task learning approach, where AttParseNet is trained simultaneously on both the segmentation and recognition tasks for 22 epochs. During this stage, the MSE loss from the segmentation task and the BCE loss from the attribute recognition task are summed to form the total loss. The

BCE loss measures the discrepancy between the predicted attribute probabilities and the ground-truth attribute labels, while the MSE loss ensures that the model maintains its ability to generate accurate segmentation masks. By optimizing both losses jointly, AttParseNet learns to capture the intricate relationships between facial attributes and their spatial localizations. We hypothesize that joint learning reduces the risk of learning spurious correlations between the occurence of facial attributes and other visual features of images which might co-occur with attributes in the training data.

We emphasize that during the validation and testing phases, we do not use the segment labels. The model's performance is evaluated solely based on its ability to predict the presence or absence of facial attributes given an input image. This approach aligns with real-world scenarios where ground-truth segmentation masks are not available during inference.

4.4.3 Baseline Model Training

The baseline model is trained using the aligned images from the CelebA training split, ensuring a fair comparison with AttParseNet. By utilizing the aligned dataset, the baseline model benefits from the implicit alignment provided by the image-level attribute labels, which serves as weak segment supervision.

The training process for the baseline model consists of a single stage, where the model is trained solely on the attribute recognition task for 22 epochs. We employ the Adam optimizer [35] with a learning rate of 1E-3 to update the network weights, leveraging its adaptive learning rate capabilities and efficient convergence properties.

During training, the BCE loss is computed based on the discrepancy between the predicted attribute probabilities and the ground-truth attribute labels. The model

learns to capture the relationships between facial attributes and their corresponding visual features present in the aligned images.

It is important to note that the baseline model shares an identical architecture with AttParseNet, including the same hyperparameters. The key difference lies in the absence of the segmentation learning task, which allows us to isolate the effects of learning localization alongside attribute recognition. By focusing exclusively on the attribute recognition task, the baseline model serves as a reference to evaluate the impact of joint learning and localization on AttParseNet's performance.

In the following sections, we present and analyze the experimental results, comparing the performance of AttParseNet with the baseline model across different evaluation metrics and datasets.

4.4.4 Experimental Setup

We implemented both the proposed AttParseNet architecture and baseline attribute classifier using PyTorch [36]. The CelebA dataset was split into training, validation, and test sets according to the provided partitions. Training was accelerated using two NVIDIA GTX-1080 TI GPUs. To prevent overfitting, we employed early stopping by monitoring the loss on the training and validation sets, stopping training when the losses became comparable.

4.4.5 Results on CelebA

The baseline model, trained on aligned CelebA images without segmentation, achieved an average attribute accuracy of 86% on the aligned test set. In comparison, AttParseNet achieved an average accuracy of 87% on the unaligned test set. While the absolute improvement is small, it is substantial considering the accuracy is averaged over 40 attributes. Figure 4.7 shows the accuracy achieved

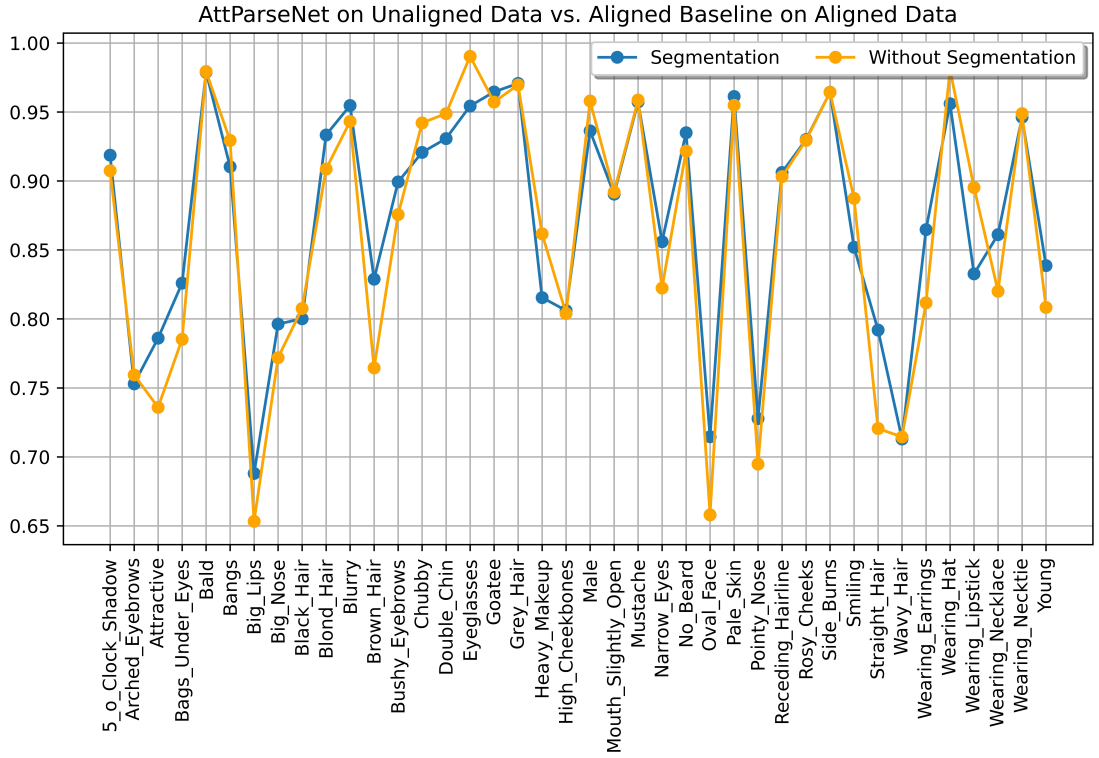


FIGURE 4.7: Average accuracy achieved on each facial attribute for the proposed architecture and a baseline model. The models are evaluated on the unaligned and aligned data sets respectively. AttParseNet is trained with the weak semantic segmentation task.

by both networks for each attribute. Table 4.1 details which attributes specifically benefited from joint training with segmentation.

Interestingly, about half of the attributes that did not improve occur in rough face segments (segments constructed from predicted landmark augmentations, see Figure 4.1). These are always on the face periphery. The lack of improvement may be due to our tighter face cropping compared to the aligned CelebA crop, reducing available contextual information for attributes like Mouth Slightly Open, Smiling, and Eyeglasses.

4.4.6 Generalization to LFWA and UMD-AED

To evaluate the generalization of AttParseNet, we tested it on the LFWA and UMD-AED datasets, both relevant for facial attribute recognition. LFWA is

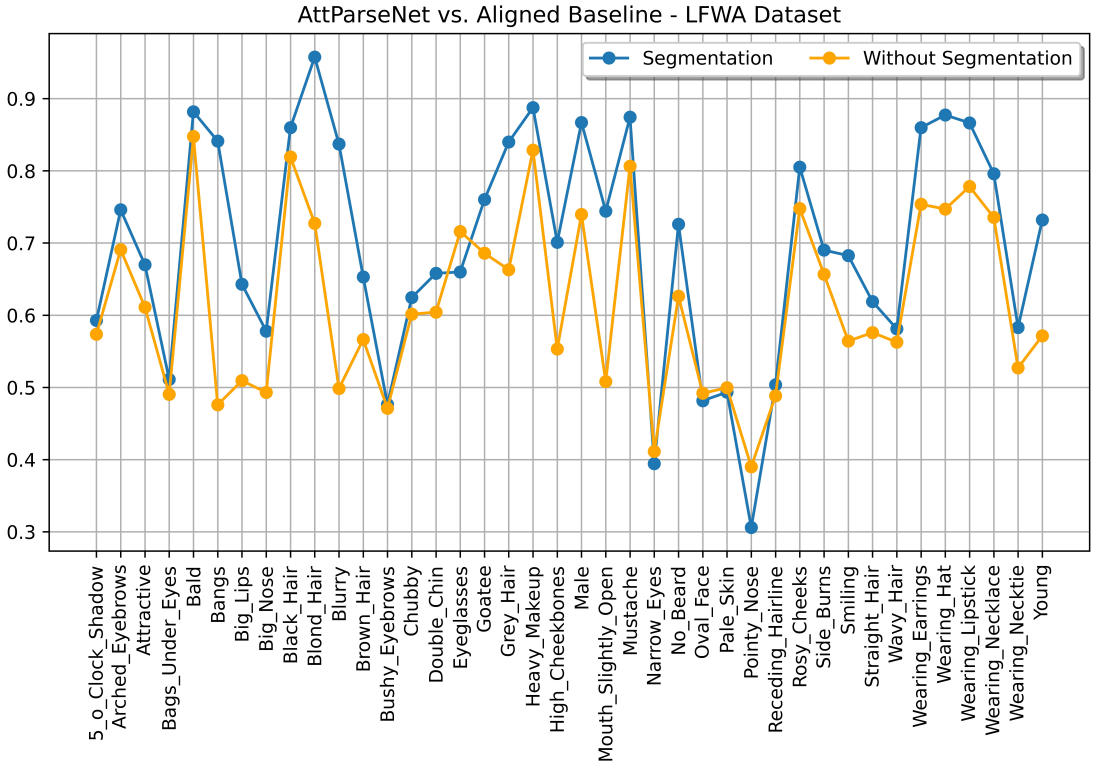


FIGURE 4.8: Average accuracy achieved on each facial attribute for the proposed architecture and a baseline model. AttParseNet is trained with the weak semantic segmentation task.

widely used, while UMD-AED has nearly perfect attribute label balance. Tests are completed by collecting predictions from AttParseNet and the baseline model for all data in each dataset, then accuracy is calculated based on the ground truth labels.

LFWA is examined first. See Figure 4.8. We note that all attribute classes show increased performance besides *Eyeglasses*, *Narrow Eyes*, *Oval Face*, *Pale Skin*, and *Pointy Nose*. The accuracy differences for the latter three were minor (¡1%), while some attributes are recognized by AttParseNet as much as 30% more accurately. *Eyeglasses* is an attribute which would benefit from an expanded segmentation mask for additional visual features.

Next, results on the UMD-AED dataset are analyzed. Accuracy for this trial is shown in Figure 4.9. Here we see improvement on all attributes besides *chubby*, *double chin*, *goatee*, *Sideburns*, *wearing necklace* and *wearing necktie*. Each of the

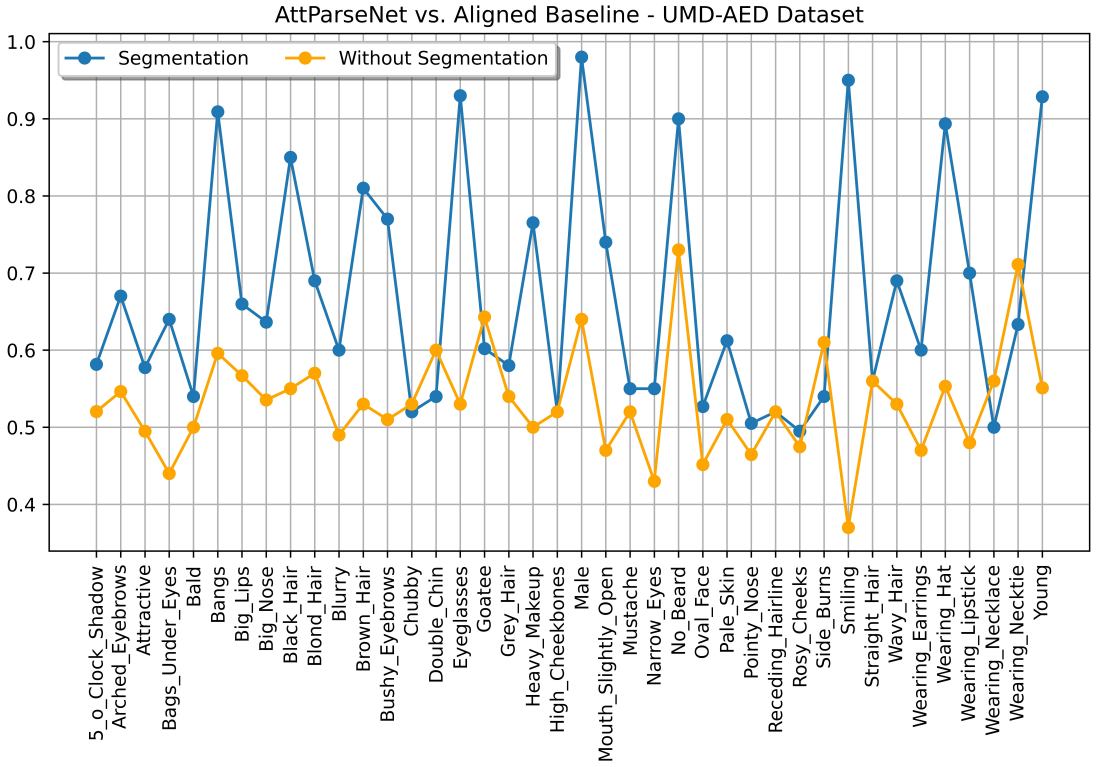


FIGURE 4.9: Average accuracy achieved on each facial attribute for the proposed architecture and a baseline model. AttParseNet is trained with the weak semantic segmentation task.

attributes that are not improved upon show less than 1% difference of accuracy, on average. This being said, AttParseNet and the aligned baseline are separated by nearly 40% accuracy for some attributes. It is of note that many of the accuracy scores for the aligned baseline classifier are within 5% of 50% accuracy score, suggesting it learned a degenerate majority-class output function for 23 attributes.

These experiments suggest that the joint learning of semantic segmentation alongside attribute classification greatly improves the performance of a base classifier. The results also suggest that enforcing features for attribute prediction coincide with the visual features which make up the target features reduces the likelihood of over-fitting, even in the presence of very few labels per attribute.

4.5 Conclusions

In this paper we introduce a new method for facial attribute recognition from images, which we call AttParseNet. Our proposed method adds weakly labeled semantic segmentation of attributes as an additional level of supervision in the attribute recognition network. We also introduce a rule-based method for generating weakly labeled facial attribute segments based on landmark points. Using these weakly labeled attribute segments we are able to add a segmentation loss to the facial attribute recognition model, in addition to the attribute recognition loss. Combining these two learning tasks in a single network results in improved facial attribute recognition and generalizability of our model on unseen data. We demonstrate the effectiveness of our method, comparing AttParseNet with the a baseline model that has the same network architecture, but is trained without the segmentation task. AttParseNet is able to take advantage of weakly labeled segmentation data to better localize and recognize facial attributes, requiring no facial landmarking at test time. In addition, there is some evidence that the semantic segmentation task has a regularization effect on the learned network weights, leading to improved model generalization to unseen data. **We emphasize that the proposed work required very little hand-labeling and no new data was collected.** Rather, we introduced a rule-based method to create weak semantic segmentation labels for added supervision in the task of attribute recognition. Future work consists of further refining the weak segmentation labels as well as a more detailed study of how attributes manifest themselves in the face.

Attribute Name	**CelebA**	**LFWA**	**UMD-AED**
5_o_Clock_Shadow	✓	✓	✓
Arched_Eyebrows	×	✓	✓
Attractive	✓	✓	✓
Bags_Under_Eyes	✓	✓	✓
Bald	×	✓	✓
Bangs	×	✓	✓
Big_Lips	✓	✓	✓
Big_Nose	✓	✓	✓
Black_Hair	×	✓	✓
Blond_Hair	✓	✓	✓
Blurry	✓	✓	✓
Brown_Hair	✓	✓	✓
Bushy_Eyebrows	✓	✓	✓
Chubby	×	✓	×
Double_Chin	×	✓	×
Eyeglasses	×	×	✓
Goatee	✓	✓	×
Gray_Hair	✓	✓	✓
Heavy_Makeup	×	✓	✓
High_Cheekbones	✓	✓	✓
Male	×	✓	✓
Mouth_Slightly_Open	×	✓	✓
Mustache	×	✓	✓
Narrow_Eyes	✓	×	✓
No_Beard	✓	✓	✓
Oval_Face	✓	×	✓
Pale_Skin	✓	×	✓
Pointy_Nose	✓	×	✓
Receding_Hairline	✓	✓	✓
Rosy_Cheeks	✓	✓	✓
Sideburns	✓	✓	×
Smiling	×	✓	✓
Straight_Hair	✓	✓	✓
Wavy_Hair	×	✓	✓
Wearing_Earrings	✓	✓	✓
Wearing_Hat	×	✓	✓
Wearing_Lipstick	×	✓	✓
Wearing_Necklace	✓	✓	×
Wearing_Necktie	×	✓	×
Young	✓	✓	✓
Total Improved	24	35	34

TABLE 4.1: Attribute improvement comparison table. A ✓ represents an improvement of average accuracy score for AttParseNet over the aligned baseline for a given attribute in a certain dataset. Conversely a × represents a lesser average accuracy score for AttParseNet.

Chapter 5

Consensus Subspace Clustering

5.1 Introduction

Supervised learning has achieved remarkable success in various domains, including biometrics and computer vision. However, the performance of supervised methods heavily relies on the availability of large-scale, labeled datasets, which can be time-consuming, expensive, and sometimes infeasible to obtain. To address this problem, unsupervised learning methods, particularly clustering, have gained significant attention as they aim to discover meaningful patterns and structures in unlabeled data [].

Traditional clustering methods, such as k-means [37], Gaussian Mixture Models (GMMs) [38, 39], and spectral clustering [40], have been applied in biometrics and computer vision. However, they often struggle to effectively group visual data. The reasons for this are two-fold. First, the image feature space is high-dimensional. This is problematic due to the curse of dimensionality [**?**], where the performance deteriorates as the dimensionality increases. Second, the visual features spatially correlate to one another. Clustering methods expect a flattened input, which removes the important spatial organization of the input data.

We posit that a mixture of traditional methods and deep learning has the potential to improve performance of unsupervised labeling of data. Recent advances in deep learning have been leveraged to overcome limitations of clustering on image datasets, with techniques such as autoencoders [41] and variational autoencoders (VAEs) [42] showing promising results in learning compact and meaningful representations of high-dimensional data without the supervision of classification labels.

The learned feature space from autoencoder architectures can be further reduced with matrix factorization techniques, such as Non-negative Matrix Factorization (NMF) [**?**]. A consensus of clustering results [**?**] can be used as a powerful technique to enhance the robustness and stability of clustering results.

Motivated by these advancements, we propose a novel multi-step clustering approach called Consensus Subspace Clustering (CSC). CSC aims to reduce the dimensionality of input data while carefully selecting the most informative features for grouping samples into meaningful clusters. By leveraging deep learning, matrix factorization, and consensus clustering techniques, CSC captures complex patterns and relationships in high-dimensional biometric and visual data.

The main contributions of this paper are:

- CSC utilizes a convolutional autoencoder and NMF to capture spatial relationships, identify informative features and flatten the input data.
- The flattened features are then passed into a VAE to extract multiple representations of the flattened data.
- Finally, consensus clustering is applied to combine clustering results from different subspaces, enhancing the stability and reliability of final cluster assignments.

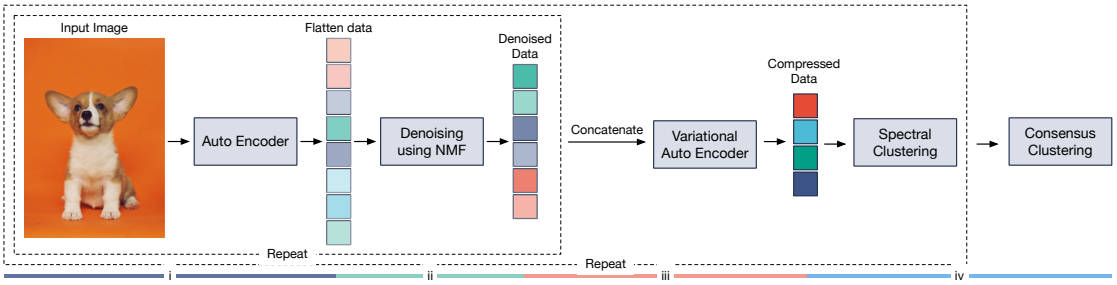


FIGURE 5.1: Overview of the proposed CSC pipeline. The method consists of four main modules: i) a flattening module using an autoencoder to extract features from input images, ii) a denoising module using NMF to remove unimportant features, iii) a compression module using VAE to generate a low-dimensional representation of denoised features and iv) a clustering module using spectral clustering to cluster images from their compressed representations.

- Experimental results demonstrate CSC's competitive performance compared to state-of-the-art clustering methods in unsupervised pseudo-labeling tasks for biometric and computer vision applications.

The remainder of this paper is organized as follows. Section **??** provides an overview of related work in biometrics, deep learning, and computer vision. Section 6.2 describes the proposed CSC method in detail. Section **??** presents the experimental setup, datasets, and evaluation metrics. Section 6.4 discusses the experimental results and compares CSC with other state-of-the-art methods. Finally, Section **??** concludes the paper and outlines future research directions in biometrics and computer vision.

5.2 Related Work

The first class of methods that we describe jointly learn to compress images into dense representations and cluster the dense representations into classes. Fard et al. [43, 44] propose methods that tune an autoencoder to generate k-means friendly representations. In [45], Xie et al. pass samples through an encoder to generate representations, cluster with k-means and correct the cluster assignments with a

clustering loss based on a KL divergence between soft assignments and their target distribution. Borrowing from Xie et al., [46] and [47] use the same learning framework with an undercomplete autoencoder to preserve the local structure of input data. Wang et al. [48] pass the input image through an orthogonal autoencoder prior to applying spectral clustering. Affeldt et al. [49] use multiple autoencoder architectures to generate multiple representations from the input data. The representations are then clustered with spectral clustering. The authors of [50] propose an architecture in which a neural network reduces the dimension of input images. The learned representations are clustered and the corresponding pseudo labels are used as supervision for training the network.

The second group of works that we highlight are miscellaneous techniques for improving cluster performance. Li et al. [51] use a boosting method to train on easier samples, then gradually expose the model to more challenging data. [52] utilizes an ensemble of classifiers to generate cluster assignments and compute a similarity graph. Finally the similarity graph is pruned to extract high confidence cluster assignments. [53] uses a modified VAE in which the latent space is sampled from a mixture of Gaussian distributions. Clustering is achieved by calculating how far the mixture distribution is from the normal distribution. Lastly, Li et al.[54] implement multi-view autoencoders for multi-view data with shared weights. Their network structure has a deep embedding clustering layer which recalculates cluster centers each iteration.

The proposed CSC differs from prior work in that the autoencoder is primarily used for flattening the data and capturing spatial relationships. In addition, related works do not filter out features of the learned representation which are not valuable for separating input data into clusters. Lastly, CSC utilizes consensus clustering to stabilize clustering results.

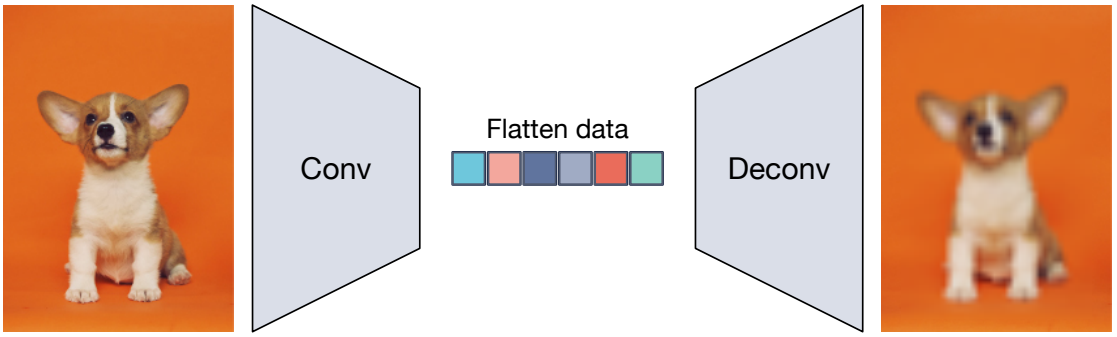


FIGURE 5.2: Feature extraction using our autoencoder. A 1-layer autoencoder is used to extract features from input images. The representation generated by the autoencoder has 500 dimensions.

5.3 Methodology

The CSC pipeline consists of four core modules, as shown in Figure 5.1. The first module extracts features from input images using an autoencoder. The second module removes noise and unimportant features by detecting meta-features with Non-negative Matrix Factorization (NMF) and inspecting reconstruction errors. CSC only retains features that significantly contribute to the reconstruction error, as these likely differentiate classes. These two modules are repeated to generate multiple denoised versions of the input. The third module is a Variational Autoencoder (VAE) that projects the denoised features into multiple lower-dimensional representations. The fourth module applies spectral clustering to these low-dimensional representations. All four modules are repeated to generate multiple cluster assignments per image. Finally, an ensemble approach determines the final cluster assignment for each image based on the assignments from each representation. The following subsections detail each module.

5.3.1 Feature Extraction

We scale pixel values in each image from 0 to 1 using min-max normalization. A 1-layer convolutional autoencoder then extracts 500 features from each normalized image via the bottleneck layer (Figure 5.2). Optimizing this model to generate a

good, compact representation requires identifying the significant visual features. We expect that the learned feature space contains some noise, which we filter out with NMF.

5.3.2 Denoising Module

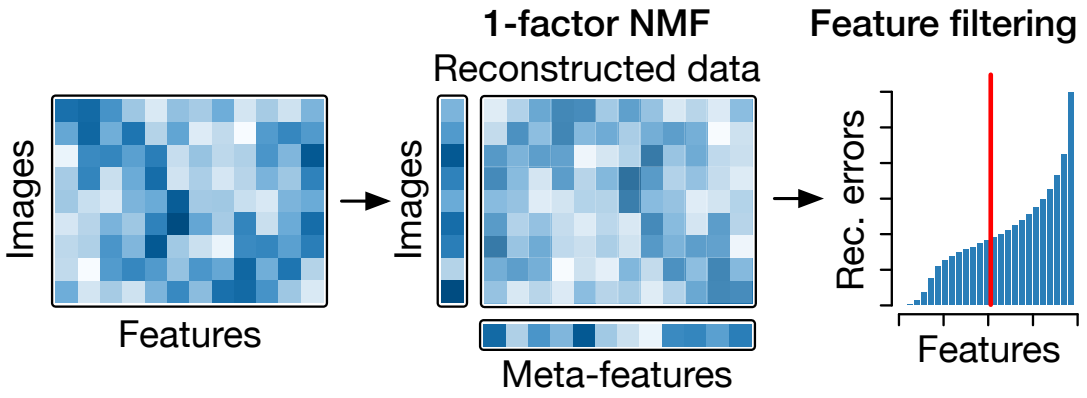


FIGURE 5.3: Denoising extracted features from input images using NMF. The original data matrix is decomposed into two vectors representing images and their features in 1-dimensional latent space. The error of the reconstructed data using these two vectors is used to rank each feature. Only 50% of features that have the largest error are kept for the next steps.

We expect that only a subset of the extracted features are useful for clustering images. Therefore, we filter out features unlikely to play a major role, using the workflow in Figure 5.3 based on 1-factor NMF. This module is represented by the following equation:

$$V_{m \times n} = W_{m \times k} \times H_{k \times n} + E_{m \times n}$$

In our system the latent vector from the flattening module is a vector V with dimensions $m \times n$, where m is the number of images and n is the dimensionality of the latent vector. NMF decomposes V into two matrices W and H which have dimensionality $m \times k$ and $k \times n$. Here, k represents the factor of the NMF model. The factors produced by NMF represent the most dominant trends in the input

vector. E is a matrix representing the error between the original vector and the reconstructed vector.

Setting the number of factors to $k = 1$ makes fitting the model difficult for features that significantly differ between clusters - the most valuable features for clustering. By attempting to reconstruct the original matrix V, we can select the most important clustering features based on those with the highest reconstruction error [55, 56]. We sort features by their absolute error and remove the 50% with the lowest error. Since the feature extraction and denoising modules are non-deterministic and sensitive to random factors, we repeat them ten times to obtain different denoised data versions, concatenating the results for the next step.

5.3.3 Variational Autoencoder

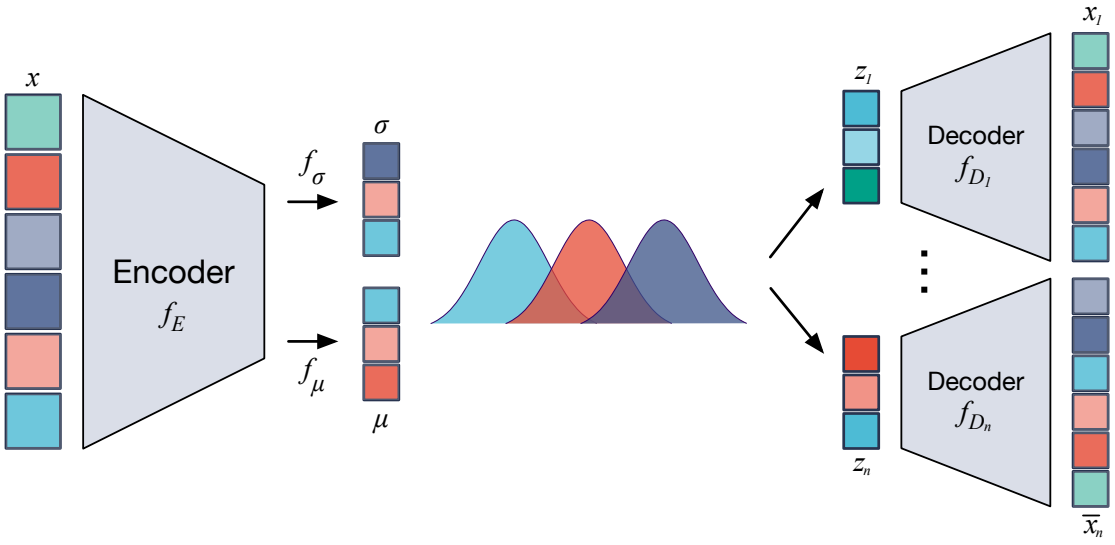


FIGURE 5.4: Compressing images using a VAE. Denoised images are compressed into multiple representations using a VAE. Multiple representations are obtained from one image. This is accomplished by adding different noise into the latent space and the use of multiple decoders to reconstruct the image. The representations of each image are used for clustering.

The previous step has removed insignificant features from the original extracted features, but the dimensions of the remaining features are still too large (2,500

features) to perform clustering efficiently. Hence, a VAE is applied to compress the significant features into a lower dimension (Figure 5.4).

The VAE architecture is similar to that of a standard autoencoder. However, rather than attempting to encode each input sample into fixed floating point features, the VAE encodes features into two vectors. These vectors represent the mean and standard deviation of a Normal distribution. This distribution is sampled to extract the latent vector. This technique results in a latent space which more smoothly transitions between classes than a traditional autoencoder.

VAEs are, however, prone to overfitting [57]. Therefore, instead of using one decoder as in a standard VAE, we use multiple decoders in our implementation to ensure that the encoder learns the generalized presentation of the input. At the end of this module we obtain three compact representations for each image by repeatedly sampling from the latent space. Output representations are gatherered into 3 groups by sample number (i.e. group 1 contains representations from the first sample of an image). These groups are referred to as subspaces.

5.3.4 Basic Subspace clustering

Spectral clustering is performed on each subspace representation to form pseudo labels for the input data (i.e., each cluster represents a class). We use spectral clustering rather than k-means to better capture non-linear relationships among images.

In our pipeline, we use the K-Means adaptation of spectral clustering, proposed by Ng et al. [58], to generate pseudo labels for input images. The clustering procedure first computes the similarity matrix for all samples to use as the input graph. It then computes the symmetric and normalized Laplacian matrix (L^{sym}). Then, the K largest eigenvectors for L^{sym}, are computed and normalized to unit length.

The eigenvectors are then used to make up the columns of a matrix. Finally, the algorithm uses K-means clustering to segment the subspace into K clusters.

To select the optimal number of clusters, we run the algorithm with a different number of clusters and select the clusters that give us the best ratio r of between-sum-of-squares and total-sum-of-squares by cluster. Since the input data can be large, for each number of clusters, we sample the input multiple times and perform clustering to obtain multiple r. We take the average of all r for each k and select the optimal number of clusters K such that r is maximized.

5.3.5 Consensus Clustering

We repeat the clustering pipeline 10 times to obtain multiple cluster assignments for each image. To generate the final cluster assignment for each image, we adopt an ensemble clustering strategy called weighted-based meta-clustering (wMetaC) [59, 60].

wMetaC uses voting from each cluster assignment to determine the final clusters. First, an image-image similarity matrix is computed, with each value representing the likelihood of two images being clustered together. Next, each image is assigned a weight by summing all pairs it appears in. These similarity matrices form a cluster-cluster similarity matrix. Finally, hierarchical clustering on this matrix selects the final clusters.

5.4 Experiments and Results

To evaluate our proposed method, we compare CSC with several existing clustering methods on two different handwritten digit datasets and one general object classification dataset. Baseline methods included in our comparison are k-means,

Deep Cluster [61], and Deep k-means [44]. The datasets used for experimentation are MNIST [62], USPS [63], and CIFAR-10 [64]. Widely used performance metrics are computed to compare CSC to baseline techniques and state-of-the-art methods.

5.4.1 Datasets

The datasets that we select for evaluation are MNIST, USPS and CIFAR-10. Each of these collections are relatively small and contain low-resolution images (32x32 pixels or less). The MNIST dataset contains a total of 70,000 images of size 28x28 (60,000 images for training and 10,000 images for testing). MNIST is relatively balanced with each of the 10 classes representing close to 10 percent of the total population. The group with most representation makes up 11.25 percent and the group with least representation makes up 9 percent. USPS contains a total of 11,000 images with of size 16x16. Both datasets have 10 classes, which correspond with the integers ranging from 0 to 9. Each image depicts a hand-written digit. USPS is mostly balanced with the largest group representing 17 percent and the smallest group representing 8 percent. The CIFAR-10 dataset contains total of 60,000 images of size 32x32x3 (50,000 images for training and 10,000 images for testing). This dataset is balanced, with 6000 images per class. CIFAR-10 provides a much more challenging task due to significantly larger feature space and diverse class labels: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

5.4.2 Methods for Comparison

Effective evaluation of CSC is achieved via comparison to state-of-the-art methods in the field. In addition, we select k-means as a baseline model. Images are flattened before being passed to k-means. K-means is run with 10 cluster centers

for a maximum of 1000 iterations or until convergence. We run k-means 20 times on each dataset and select the run with best results for comparison. The selected state-of-the-art methods are Deep Cluster [61] and Deep k-means [44]. Results shown in Table 5.1 are those reported in each publication.

5.4.3 Metrics

We use Accuracy (ACC) and Normalized Mutual Information (NMI) as metrics to evaluate performance of each method. Accuracy and NMI metrics are used to be consistent with the evaluations in the original papers of corresponding methods included in the comparison. The metrics are calculated as follow:

$$ACC = max_m \frac{\sum_{n=1}^{N} \mathbf{1}(l_i = m(c_i))}{N}$$

where $\mathbf{1}(.)$ is an indicator function, l_i is the true label, c_i is the label assigned by the clustering method and $m(.)$ denotes all possible one-to-one mappings between clusters.

$$NMI = \frac{I(l, \mathbf{c})}{(H(\mathbf{l}) + H(\mathbf{c}))/2}$$

where $\mathbf{l}$ denotes the ground truth labels, $\mathbf{c}$ is the cluster assignments, $I(.)$ is the mutual information metric, and $H(.)$ is the entropy.

5.4.4 Results

Table 5.1 shows the Accuracy and NMI for CSC and comparison methods on the MNIST, USPS and CIFAR-10 datasets. On the MNIST task, CSC far exceeds performance of the baseline and outperforms the other methods in accuracy. Deep Cluster reports slightly better NMI for MNIST and Deep K-means outperforms CSC in both metrics on the USPS dataset. In the case of Deep Cluster, the margin

of difference is very slight and shows that CSC is competitive with state-of-the-art on this task. Regarding Deep k-means, we believe that the architecture is better suited for the smaller feature space found in USPS. Each image in this dataset contains a total of only 256 features. To reinforce this claim, we point to the method's decreased performance on the larger MNIST and CIFAR-10 datasets. We note that the authors of Deep Cluster and Deep K-means did not evaluate their methods on the CIFAR-10 dataset.

TABLE 5.1: Performance of K-means, Deep Cluster, Deep K-means, and CSC on MNIST, USPS and CIFAR-10 datasets.

Method	MNIST		USPS		CIFAR-10	
	ACC	NMI	ACC	NMI	ACC	NMI
K-means	0.58	0.49	0.48	0.42	0.14	**0.12**
Deep Cluster	**0.86**	**0.83**	0.67	0.69	—	—
Deep K-means	0.84	0.80	0.76	0.78	—	—
CSC No Flatten	0.85	0.79	0.83	0.78	0.12	0.08
CSC No Filter	0.83	0.76	**0.84**	**0.79**	0.14	0.10
CSC No Voting	0.82	0.77	0.82	0.76	0.14	0.10
CSC	**0.86**	0.81	0.83	**0.79**	**0.15**	0.11

Complete analysis of CSC requires an understanding of how each component in the pipeline effects the end performance of the model. Referencing the latter half of Table 5.1, removing the flattening module reports the least change out of all modules. However, flattening appears to become more important as the complexity of the dataset increases. Next, the filtering module is particularly important for MNIST, but less important for USPS. This is likely because the samples in USPS are mostly separated before being processed by the VAE, see Figures 5.5 and 5.6. Last, voting or consensus clustering is very important for stability of clustering results. In our trials without voting, results can be extremely variable.

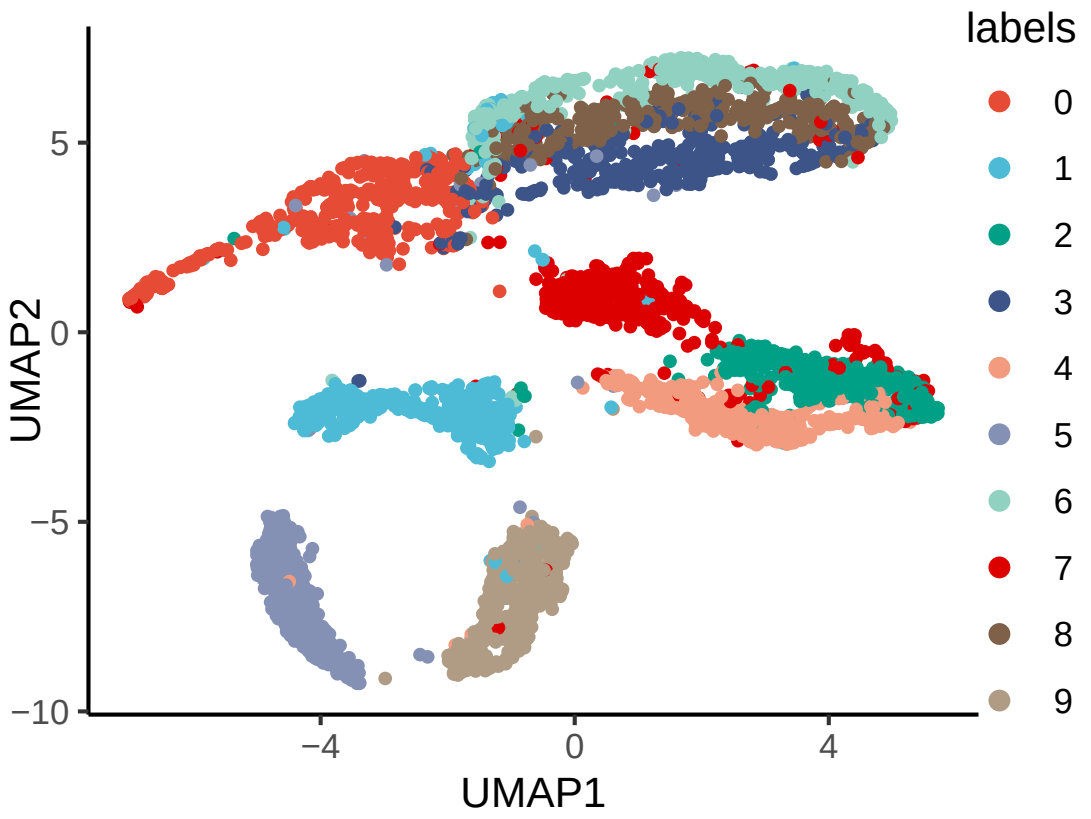


FIGURE 5.5: A UMAP [3] visualization of the raw USPS dataset. Each colored dot represents an input sample.

5.5 Conclusion

In this work, we have introduced a novel method for providing pseudo labels on arbitrary image data, which we call CSC. To the best of our knowledge we are the first to present a deep clustering method which removes inconsequential features from input data and learns multiple representations of the data to reinforce the robustness of selected cluster labels. Our experimentation shows that our work is competitive with, and in some cases, exceeds the state-of-the-art for deep clustering of image data. Future work in this area could introduce a confidence measure to the sample in each cluster. Additionally the method could be expanded to process data beyond images.

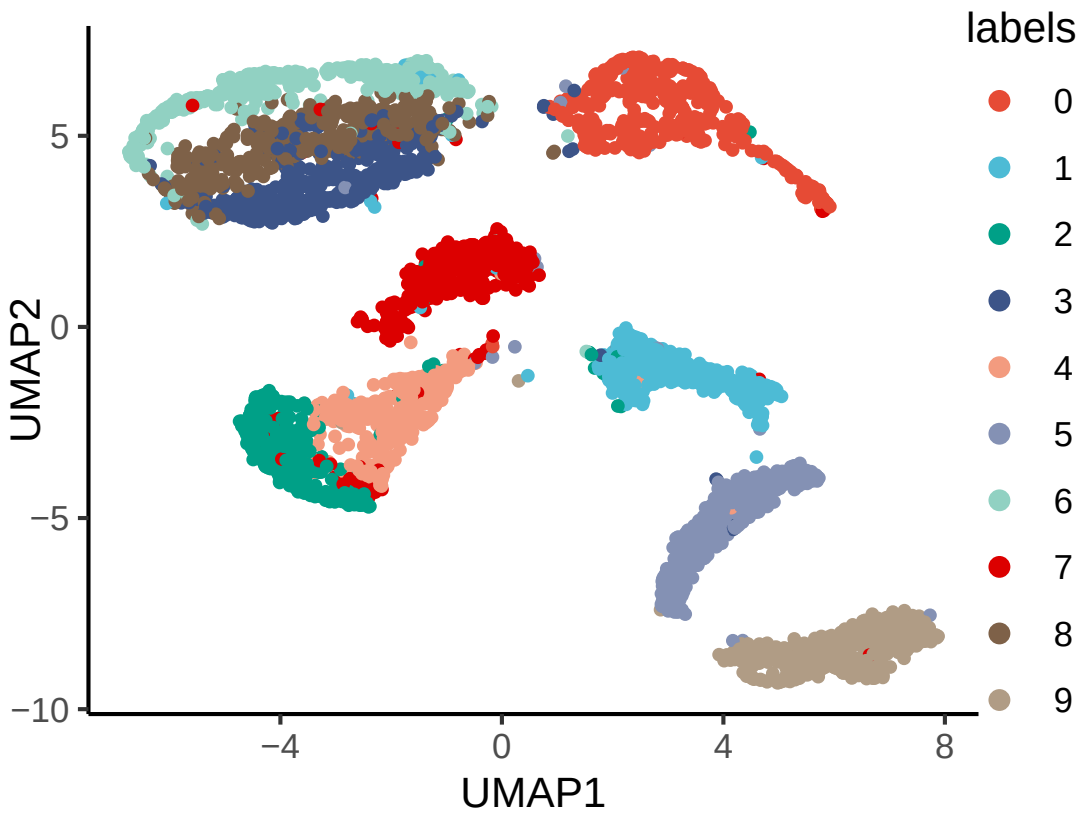


FIGURE 5.6: A UMAP [3] visualization of the USPS dataset after it is processed with CSC. Each colored dot represents a latent representation of an input sample from the dataset. Note that the points within each cluster tighten together and the clusters are separated by a greater margin than those that appear in Figure 5.5.

Chapter 6

Deep Vision Model Perception of Gender From Faces

In recent years, deep neural networks (DNNs) have achieved remarkable performance within a wide range of application fields. Yet, the artificial intelligence surge appeared in conjunction with an increase in model complexity [65]. While the advancement of intelligent systems can be construed as positive, it has opened the door for a new set of challenges. How does a model make decisions? What information is most important to a model? How can model perception be described in a way human users can understand? And if I do not understand what a model is learning, can I trust it? Although powerful in terms of predictions and results, AI algorithms suffer from issues of opacity, that is a difficulty gaining insight into their internal mechanisms. This further compounds the issue of presenting critical tasks to a system that cannot explain itself [66]. Explainable AI remains a relatively open problem.

We present an architecture-agnostic approach for illuminating black box models (e.g. CNNs). We derive explanations for a state-of-the-art convolutional neural network (CNN), ResNet-50, when tasked with the problem of gender classification. If the goal is to build intelligent agents capable of providing explanations to

people, then the logical starting point is to mimic how humans explain decisions to each other [67]. We apply frameworks developed in neuroscience to explain model decisions, and argue that the resulting interpretations are the most natural and expressive from the standpoint of human understanding. Our proposed method emulates the most widely utilized experiment in cognitive science for interpreting decisions: testing the reaction of a subject to carefully prepared variations of the same data [68]. We analyze the variance in evaluation metrics resulting from exposing ResNet-50 to occlusions to determine what information is most crucial to the model. As a byproduct, we extract key regions of faces from the large-scale, unconstrained dataset CelebA [69]. The structure of our approach allows for the immediate understanding of model attention, expressed in non-technical terms directly comparable to human perception.

We compare our work to prior interpretability methods such as CAM[70] and Grad-CAM [71], showing that the results are simpler to generate, less susceptible to interference, and more globally descriptive. We further demonstrate the possibility of combining our method with popular existing work, such as heat-mapping, to generate visualizable local explanations.

6.1 Related Work

There are large bodies of work on model interpretability and gender discrimination in the fields of artificial intelligence, psychology and neuroscience. We review the relevant literature here.

6.1.1 Model Interpretability

The previous decade has been witness to an explosion in the use of ubiquitous opaque decision systems, specifically DNNs. Artificial intelligence applications

now exist in such critical areas as medicine, security and finance. Additionally, software libraries such as Tensorflow [72] and PyTorch [73] have made machine learning techniques widely accessible. Sophisticated models are now commonly applied in both research and industry. However, as the population of models evolves in complexity, so too must the ability to interpret them. The lack of explanation associated with black box DNN systems, that is systems whose internal logic is not visible to the user, constitutes both a practical and ethical issue.

Adadi et al. posit four primary motivations driving explainable AI (XAI): justification, control, improvement and discovery [66]. So, with the need for interpretable artificial intelligence clearly established, we must first address the question: at what point can a system be classified as understandable? Quasi-mathematical definitions of interpretability abound within the machine learning community, however we lack a formal technical meaning at present. Taxonomic work suggests that an interpretable model can be determined by (1) transparency, i.e. how does the model work? and (2) post-hoc explanation, i.e. what more can the model tell us? [74]. So, in the broadest sense, we say that a model is explainable if its decisions are amenable to human replication and quantification. We exploit this desired consistency with human intuition to assemble machine justifications that can be closely compared to observed human decision processes.

Previous work in interpretability can be distinguished, at a very high level, between reverse engineering and design of explanations. Given a dataset of training decision records, the former technique reconstructs an explanation for a decision, while the latter develops an interpretable predictor model alongside the decision set [75]. There is also a fairly popular third method: forcefully simplifying a model until it falls within the small class of recognizable explainable systems. However, this technique does not contribute to the development of XAI, almost always necessitates a loss of accuracy, and sacrifices performance for the sake of simplicity.

Considerably more robust reverse engineering methods are widely used; key examples specific to CNN architectures include visualizations such as heat maps and bounding boxes [74].

Heat mapping techniques generate a localization map highlighting the important regions within an input image for discrimination. Gradient based methods, such as DeConvNet [76] and Guided Backprop [77] [78] backpropagate the gradients for a class label to the image layer, while network-activation methods such as CAM and Grad-CAM visualize the linear combination of a late layer's activations and class-specific weights [79] [80] [81] [71]. These constructions, while useful and aesthetically gratifying, are still heuristic notions of image saliency [82]. Furthermore, heat-maps are isolated by example, hence unable to provide overarching generalizations of a model's attention. Unlike these methods, the proposed method is able to generalize model behaviors, notably over the entirety of a dataset.

Instances of design techniques, such as LIME and SHAP, explain the predictions of a classifier by learning a comprehensible sparse local predictor around the decision [83] [84]. Yet, [85] demonstrates that such post-hoc explanation techniques relying on input perturbations are unreliable, and can be easily fooled into providing innocuous explanations unreflective of underlying biases. Our method derives a set of rules explaining the logic motivating the black box architecture, thus providing explanation at a global level.

6.1.2 Gender Recognition

6.1.2.1 Automated

Gender classification using facial data is a pattern recognition task which follows no simple algorithm, and is easily modifiable according to data quality and scenario (e.g. age, makeup, ethnicity, lighting, etc). Early neural networks (NNs) and

SVMs approached the problem using pixel intensity values as classifier input [86]. As a preprocessing step, dimensionality reduction methods such as Principal Component Analysis (PCA) were popularly used in early studies to reduce an image vector to a representation in lower-dimensional space [87]. Later work by Cottrell [88] used an autoencoder network to extract reduced, whole-face features (dubbed 'holons'), which were given to small neural networks trained to classify gender. Ng et al. categorized feature extraction for gender recognition into appearance-based and geometric-based methods: the former works by operation or transformation of image pixels, while the latter uses fiducial points marking features such as the nose and mouth [89].

With the advent of deep neural networks, gender recognition in automated systems has achieved near human performance [89].

6.1.2.2 Behaviors in Humans

Gender categorization has received considerable attention in the fields of psychology and neuroscience, with particular attention given to determining the contribution of various visual cues. Cues refers to sources of information along a given dimension, and are typically considered as either shape or surface [90]. Examples of surface characteristics are color, brightness and texture, while shape cues are drawn from an object's shape and positioning. In a neuroscientific context, the most widely held hypotheses support three general observable properties:

1. Shape cues, both in-plane and three-dimensional:
    (a) provide significant information about gender which human observers are able to exploit; and
    (b) are more heavily weighted over surface cues, and thus form the basis for delineating constituent facial parts [90].

2. Face recognition is typically cued by the diagnosticity of distinct local features such as the eyes and mouth [91].

3. Face perception is normally characterized by a unique processing style emphasizing holistic or configural aspects of the face over its specific features [92] [93] [94], but is disrupted by missing information and altered to a more feature-based process.

Our proposed method mimics the characteristics of human perception in order to create model explanations that are consistent with our previous definition of interpretability, i.e., replicability by a human. We do so by testing the model's response to the shape cues of well-defined facial regions (e.g. the mouth, nose) using occlusion. When facial information is missing, humans switch between the aspects they rely on for gender recognition: the information prioritized in each scenario forms the basis for understanding human perception as a whole. Similarly, for each instance of missing data, we determine which cues are high-priority to the model, and use the metric responses to specific features as grounds for performance trends.

6.2 Proposed Method

In this section, we define relevant background terminology and present the proposed method.

6.2.1 Defining Image Occlusions

A black box classification model $f : X \rightarrow Y$ is a map from the feature space to the decision space obtained through a nontransparent learning process. To solidify the discussion, consider as input the set of RGB images $X = \{x \in X | x : \Gamma \rightarrow R^3$ s.t.

$\Gamma = \{1, ..., H\} \times \{1, ..., W\}$ is a discrete domain}. In the case of n possible decision classes, the output space is defined as $Y = \{y \in Y | y \in R^n, y_i$ is the probability score of class i, $i \in [1, n]\}$.

We are interested in finding image sections that impact the model's (f) output; in other words, given an input image x, saliency is achieved by determining which areas of x are used by f to produce the output $f(x)$ [82]. We can do so by "deleting" regions Υ of x, and measuring the corresponding changes in $f(x)$, thereby characterizing the relationship between the perturbation and the output. Although this proposal is conceptually simple, it is not without complication.

First, we must specify what it means to "delete" information. We wish to simulate a naturalistic imaging effect in order to lead to more meaningful perturbations and hence explanations. Since we do not have access to the image collection process, the most immediate proxy is to replace a region Υ with a constant value. Formally, we introduce a mask $m : \Gamma \rightarrow \{0, 1\}$ associating each pixel u in an input image x with a scalar value $m(u)$. $x(u)$ is the pixel's original value in the image. The occlusion operator can then be defined as:

$$[g_m(x)](u) = m(u)x(u) + (1 - m(u))c \tag{6.1}$$

where c is the average replacement color value. g allows for the replacement of the region Υ with any desired pixel value, for example the average skin color value from the image x.
In the context of this study, we choose $c = 0$, i.e. replacement with the color black; referred to as occlusion.

The map m describes the region Υ to be occluded: for each pixel u, we specify if u will be preserved or removed. If $m(u) = 1$, the value of g is $u's$ original color value

in the image, $x(u)$, so the pixel will remain unchanged. If $m(u) = 0$, g evaluates to c, so the pixel u will be filled with the replacement color value.

Compiling an explanatory rule for a black box system requires first specifying which variations of the input image x will be utilized in studying f. As our definition of model interpretability emphasizes a communicable relationship with human face processing, we select occlusion regions commonly proctored in human perception studies and natural settings [91]. In other words, we look within the set of highly influential human gender recognition regions for perturbations which maximally impact model attention.

6.2.1.1 Meaningful Perturbations

Intuitively, we wish to search for deletion regions that are informative, i.e. ones that cause the target score for predicting a given class to decrease significantly. Gender discrimination is a two class problem, but in a more general context, if x is an input image and f_c (the probability of target class c) is the cth component of the output vector $f(x)$, a mask m that occludes a key region would result in the hypothesis score dropping substantially: $f_c(g_m(x)) << f_c(x)$. We stipulate an extension of this principle: rather than relying on class probability as an indicator of a meaningful perturbation, we utilize accuracy, precision, recall and F1 score. The latter three metrics indicate model bias and potential class-skewing when data is obscured (i.e. which regions lead to which class prediction).

6.2.2 Methods

We initially train a classifier on a face dataset for the problem of gender prediction. Next, we systematically remove key image regions, and use these occlusions as testing data. If p is the prediction accuracy of the network when testing on the

original images, and p_m is the accuracy when testing on the occlusion images given by the map m, then we let $q = |p - p_m|$. Equivalently, q is the decrease in accuracy when the region given by m is removed.

We identify an image region Υ that is conducive to large variations in output decisions via a sorted, descending list of q values, termed 'q-list'. Next we reverse the process: training with only Υ visible in the image and testing on the original unmasked image. The first direction asks how well the model can classify a full face with a region absent; i.e., how does the region contribute to model discrimination, given the presence of all other facial regions? This is analog to the neuroscientific concept of structural (or configural) importance. Conversely, the second direction in the methodology tests the model's ability to accurately determine if an isolated region is male or female. In other words, it checks the information encoded/perceived within a single region; a characteristic referred to as featural importance in human studies. Figure 6.1 shows a diagram of the two evaluation scenarios. The dual directions serve to eliminate spurious correlations and local explanations that are not robust to artifacts.

We adopt the language of structural and featural importance accordingly. A region with high structural importance will produce lower classification accuracies when it is isolated and occluded. A region with high featural importance will itself be predicted male or female with higher accuracy than a non-featurally important region. We note that since featurally important regions show strong distinctions between male and female samples, and resemble each other within the same group, they exhibit high intraclass similarity.

6.2.2.1 Evaluation

We further examine changes in accuracy, precision, recall and F1 score between occlusions to extrapolate model performance trends. As our method primarily

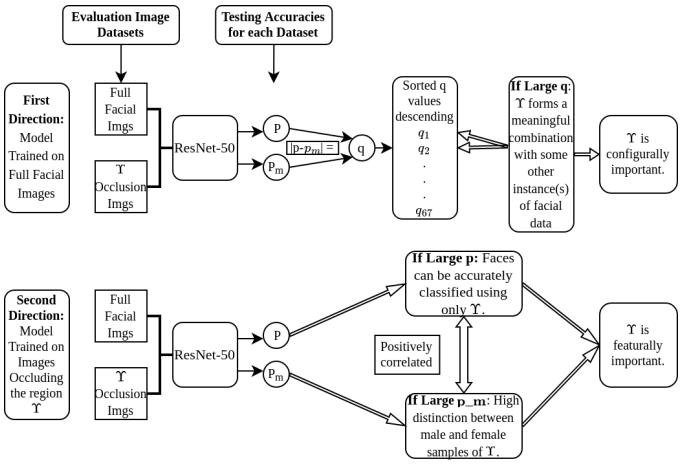


FIGURE 6.1: **The Two Directions of Evaluation**.
The figure shows the testing phase for two distinct network instances: the first row corresponds to when ResNet-50 is trained on full facial images, and the second to when training is done on a set of occlusion images of a given region Υ. The test accuracy for the full facial dataset, regardless of training scenario, is referred to as p. Similarly, the accuracy for occlusion images is always denoted p_m. In the first scenario, the absolute difference between these values defines the relationship between model output and the region Υ, a quantity referred to as q. q-values are sorted into descending order: the higher the ranking on the list, the more powerfully Υ combines with existing regions and contributes towards accuracy, a measurement know as configural importance. In the second scenario, p indicates how accurately faces can be classified when the region Υ is the only learnable information, i.e. its individual contribution to model performance. p_m determines how distinct samples of the region appear between classes. Clearly these two metrics are highly interrelated: for example, if mouths are found to be easily identifiable as male or female, and also form the only basis for classification, then p and p_m will always both have large value. High values indicate that Υ is featurally importance.

seeks to express decision justifications with transparent human vocabulary, we use extensive analysis to generalize understandable patterns in network behavior. A behavior that has been observed and validated in experimentation is referred to as a trend. Our trends comprise a set of rules describing the logic behind the black box model, and so provide interpretability at a global level.

Our approach does not commit to finding a single representative perturbation, so does not run the risk of triggering artifacts of the black box model. Thus, we avoid the pitfall experienced by other local explanation techniques [95]: identifying particular inputs that drive the model towards nonsensical or unexpected predictions.

Powerful explanations should generalize to the greatest extent possible, thus we test our trends with 5-fold cross validation on both high (224x224) and low (32x32) resolution datasets.

Many interpretability techniques suffer from a lack of applicability; our proposed method is both model-agnostic and scalable. Its verifiable nature (i.e. direct modifications to an image) and quantitative criterion (i.e. maximally altering returned metric scores) allows us to compare proposed saliency explanations showing that those produced by our model are more informative.

6.3 Experiments

6.3.1 Model

This work attempts to interpret the decisions of one of the most popular state-of-the-art architectures: ResNet-50 [96], [**?**]. We use the PyTorch library [**?**] for implementation, training, and testing. Weights and Biases [97] is used to sweep for optimal hyperparameters (w.r.t. minimizing validation loss) in the cases of both high and low resolution datasets. Table 6.1 presents the results.

TABLE 6.1: *ResNet-50 Model Hyperparameters Found Using a Parameter Sweep with Weights and Biases*

Field	Low Res. Model	High Res. Model
Image Size	32x32	224x224
Batch Size	80	138
Learning Rate	0.0005786	0.001033
Epochs	40	54
Momentum	0.4354	0.5091
Dropout	0.5091	0.3219

Note that the learning rate is updated during training with the Adam optimizer. The value in the table is parameter's starting value.

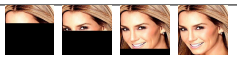
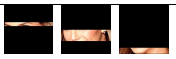
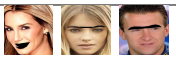
6.3.2 Data

Our experiments utilize the challenging, publicly available dataset, CelebA [69]. Originally collected for attribute classification, CelebA contains roughly 200,000 images (split roughly 8:1:1 for training, validation and testing) which vary widely with regard to subject pose, illumination and image quality. The class balance of CelebA is 42% male and 58% female. We want to mimic the settings for cognitive experiments in order to be able compare machine and and perceptions. So, we define five primary facial regions consistent with cue-driven human perception [68], and index accordingly:

1. Eyebrows
2. Eyes
3. Nose
4. Mouth
5. Chin

Coordinates for these five sections are included in the CelebA data repository. For more precise isolations, we extracted 68 landmarks from each sample image using a face detector [98]. To increase the robustness of local explanations, we consider an important region within the context of its surroundings: we include analyses of a feature (e.g. only the nose), the horizontal extension of the feature (e.g., the nose and cheeks, from ear to ear) and distinct combinations of the prior. Furthermore, we test for symmetrical and axis-distributed information encoding by systematically removing horizontal and vertical image data. We generate a total of 67 variations of the CelebA dataset. Descriptions of all datasets can be found in Table 6.2.

TABLE 6.2: Samples from each of the occlusion datasets used in analysis (67 total).

Dataset	Occlusion	Examples
Right_to_left *Segments 1 - 7*	The segments successively occlude increasing vertical image percentages in increments of 1/7 the image size, moving from right to left.	
Left_to_Right *Segments 1 - 7*	The segments successively occlude increasing vertical image percentages in increments of 1/7 the image size, moving from left to right.	
Top_to_bottom *Segments 1 - 5*	The segments consecutively reveal horizontal strips surrounding facial regions, moving from top to bottom.	
Bottom_to_top *Segments 1 - 5*	The segments consecutively reveal horizontal strips surrounding facial regions, moving from bottom to top.	
Permutation_blackout_pairs *Segments 1 - 10*	The segments remove every possible unique combination of two distinct facial regions, along with their immediate horizontal surroundings.	
Permutation_blackout_triples *Segments 1 - 10*	The segments remove every possible unique combination of three distinct facial regions, along with their immediate horizontal surroundings.	
Just_region *Segments 1 - 5*	Each of the five segments retains only the horizontal strip containing a single facial region.	
Region_blackout *Segments 1 - 5*	Each of the five segments removes only one horizontally extended facial region.	
Just_region_contoured *Segments 1 - 5*	Each of the five segments display only the contoured facial region without any surrounding facial information (e.g. the eyebrows absent the forehead.)	
Region_blackout_contoured *Segments 1 - 5*	Each of the five segments removes one contoured facial region (e.g. occluding the nose while preserving the cheeks.)	

6.4 Results

We experiment on both low (32x32) and high (224x224) resolution versions of CelebA. Significant results from each are briefly reviewed in the following sections. Full metric tables will be available on our Github upon publication.

6.4.1 Low Resolution Image Data

Resnet-50 achieves 99.99% gender prediction accuracy on low-resolution, full-facial images. We generate our q-list, a sorted descending list of changes in testing accuracies, corresponding to when the full facial image is visible to when the region given by the map m is occluded (the first direction in our method). Our q-list quantitatively characterizes the relationship between the input perturbation defined by deleting an isolated region, and the performance of the model. The list indicates that the ranking of regions, in order of greatest to least effect on classification accuracy, is: Nose, Mouth, Eyebrows, Eyes, Chin. Testing the model's ability to predict the gender of only an individual region (the second direction in our method) yields an interesting polarity. The ranking, from most to least significant, is: Eyes, Eyebrows, Mouth, Chin, Nose. These two lists illustrate the difference between the predictive effect of a missing region, and the intraclass similarity of that region. That is, the former describes the region's configural importance, while the latter describes its featural. For example, the first direction of the method shows that the nose is the most configurally and least featurally important of all regions. This implies that the nose region enhances the effect other facial elements have on performance, and is more powerful when combining with other regions than when being used by itself as a basis for gender prediction, (as it cannot be particularly distinguished as male or female).

The low resolution model relies heavily on the mouth for discrimination. This region, when occluded along with any one or two others, produces lower classification accuracies than any other combinations of two or three regions. For example, images with two regions occluded, where one is the mouth, are classified with an average of only 49% accuracy. In contrast, the average classification accuracy of images with any two regions except for the mouth occluded is 73.5%. Indeed, as long as the mouth is preserved, we are able to remove up to three facial regions and still outperform 80% of the permutation blackouts that remove only two. Metrics demonstrate over a 26% increase in accuracy when providing lower (nose and below) as opposed to upper (strictly above the nose) facial data. When taken together, these behaviors show that the model prioritizes lower facial data when evaluated on low resolution images.

6.4.2 High Resolution Image Data

With regard to high resolution data, ResNet-50 once again reaches near perfect (99.97%) accuracy for gender prediction. However, the introduction of high resolution images causes the model to shift focus from the lower to the upper face. The q-list generated from individual region occlusions reflects the expected change in high-priority regions. From most to least, the order is: Eyes, Nose, Eyebrows, Mouth, Chin. Yet, the ranking of region intraclass similarities remains entirely unchanged from the 32x32 case: Eyes, Eyebrows, Nose, Mouth, Chin. Hence, for each region, the featural contribution is consistent between high and low resolution, and clustering within the male and female classes is preserved. The diagnosticity of an individual region is not changing, but rather the weight attached to it by the model when processing faces overall. Therefore, we are able to clearly observe a redirection in model attention towards the upper face.

6.4.3 Generalizable Behaviors

Trends that uniformly hold across both high and low resolution data are presented below.

1. *Vertical vs Horizontal*: We find that classification accuracy decreases consistently when information is removed vertically, but aggressively when deleted horizontally by landmark. Vertical cross-sections of the face are generally only informative when at least 57% of the face is shown; that is, once slightly over half of the face is vertically removed, prediction accuracy decreases to and stays at a global minimum. However, for removals that occur before this threshold of information loss (i.e. more than 57% of the face shown), accuracy steadily decreases in average increments of 5.5% with each occlusion. Removing horizontal cross-sections consecutively (1) causes stronger decreases in accuracy (an average of 15%), and (2) does not result in the same uniform decrease seen with vertical removals.

2. *Spatial Distribution of Information:* When occluding vertical regions of the face from right to left, or left to right, performance changes symmetrically. That is, scores corresponding to occlusions of the right side of the face match scores corresponding to occlusions of the left side of the face. Accuracy will be affected in the same way regardless of which side is removed. This property is not observed when deletion regions shift horizontally from bottom to top or from top to bottom. Removing regions of the upper vs lower face yields largely different performance metrics. Indeed, each horizontal removal corresponds to a unique landmarked region, and causes a distinct change in accuracy.

   By contrast, vertical removals follow a loose spatial pattern, that is, the quantity of raw facial data in a vertical section positively correlates with classification accuracy. Segments which are closer to the center of the face,

and consequently contain larger portions of it, are more powerful for prediction. For example, the two vertical segments splitting the centerline of the face are responsible for more than a 50% increase in prediction accuracy when present vs not present. The division extending horizontally across the nose and cheeks (Just_region_segment3, see Table 6.2) holds the most facial data of any isolated region, but is not the most informative. In other words, the potency of a vertical cross-section is given by the density of facial features within it, while the importance of a horizontal component is determined by the q-list ranking of the contained region.

3. *Classification Bias:* Classifying images with three or more upper facial regions occluded results in low precision (around 0.4), and high recall (around 0.9), indicating strong model bias toward the male class when information is removed. When classifying the lower vs upper face, ResNet-50 switches from a very high to a very low recall. More precisely, the appearance of information contained in the nose/cheeks shifts the balance of class predictions, and results in the appearance of significantly more positively-labeled (male) samples.

   In other words, the upper face is predicted female with extreme bias, while the lower face is frequently categorized as male.

4. *Highly Influential Regions:* In either situation of low or high resolution, one region is heavily prioritized for discrimination (the mouth in the former case, eyes in the latter). By simply identifying this feature, we are already able to reconstruct model decisions with good accuracy (an average minimum threshold of 39% accuracy with the worst-performing combination of this region and one other, and average maximum of 74% accuracy with its best-performing combination.) Figure 6.2 demonstrates the change in powerful image regions across resolutions.

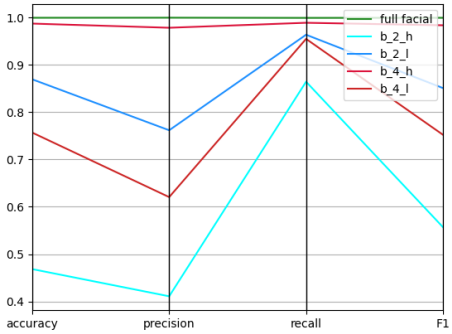


FIGURE 6.2: Parallel axis graphs of performance metrics when testing on Region_blackout datasets 2 (blue/turquoise lines) and 4 (pink/red lines), which remove the eyes and mouth respectively. Label notation is dataset_resolution; for example, b_2_h describes the performance of the high resolution model on Region_blackout 2, while b_4_l is associated with the scores of the low resolution model on Region_blackout 4. The green line represents the baseline model performance on full facial images, with uniformly high scores across all metrics. It is apparent that the high resolution model performs extremely poorly on images occluding the eye region in contrast to the low. Similarly, the low resolution model is highly affected when the mouth is removed, but the high resolution model shows almost no change in performance from the baseline.

These trends describe high impact factors on the decision processes of ResNet-50. Potential applications are diverse. For example, we found that the mouth and chin regions are of low priority to the high resolution model. If we then eliminate these sections, we are able to discard nearly 40% of the image data and still maintain 98% classification accuracy. According to the q-list associated with 224x224 datasets, the extended horizontal eye region (which on average covers only 18% of the face), is highly discriminative. Training with only this section visible we are able reach 77% accuracy, significantly outperforming any previous facial isolations. Similarly, attempting to classify images with just the eye region occluded reduces accuracy to 46%, a level lower than that of random guessing. So, in space-saving cases where some facial data must be eliminated, we can (1) avoid removing powerful regions, and (2) achieve any specified accuracy using only the smallest possible subset of facial data.

Existing local interpretability techniques often require impractically large numbers of input perturbations to be tested. The proposed method simultaneously:

1. Finds highly influential decision regions.

2. Explicitly measures the sensitivity of the model to each region.

3. Preserves comparative speed and simplicity.

Furthermore, since our metrics are averaged over the entirety of the CelebA dataset, we avoid the pitfalls of noisy input and chance correlation found in methods that analyze by example. Thus, we demonstrate the effectiveness of the proposed featural scheme.

6.4.4 Incorporating Heat Maps

For the sake of comparison, we use the CAM method [70] to generate saliency maps of sample images. An example is displayed in Figure 6.3. Features which are highly weighted by the model in the final convolution layer, and thus considered influential for discrimination, appear red in the corresponding visualizations.

The simplicity of our method allows for seamless combination with many other techniques. The proposed method integrates with heat-mapping to yield a visual representation of how the model shifts attention based on the information available.

The saliency masks shown in Figure 6.4 support many of our evaluations from sections 5.2 and 5.3. The third and fourth images in each row show alternate portions of the face occluded, but with the eyes still visible. In these cases, the

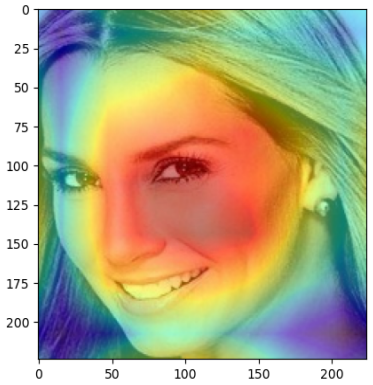


FIGURE 6.3: *Example heat map generated by ResNet-50, trained on 224 × 224 images. The red regions are highly weighted by the model (in the final convolution layer) during classification. This map illustrates a strong model attention towards the eyes and horizontal extension of the nose, in the right vertical side of the face. Best viewed in color.*

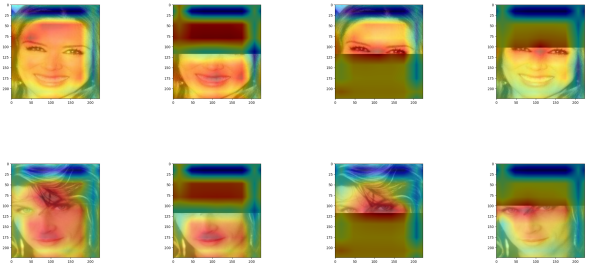


FIGURE 6.4: *224 × 224 heat maps corresponding to test images containing (from left to right) the full face, the bottom half, the top half, and the eyes and bottom half. The maps show the redirection of model attention to new regions when previously prioritized information is no longer available. For example, the third column shows that when the mouth is removed, the eyes are used almost solely for gender prediction. When they are no longer visible, as in the second column, more attention is given to the mouth. The fourth column indicates that by removing the eyebrows, it is possible to focus the key area of the image to the eyes even more precisely. Best viewed in color.*

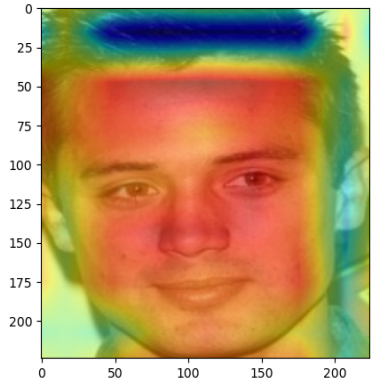


FIGURE 6.5: *A non-informative heat map. This map is essentially useless for determining key areas of an image. It was extracted from an example set of 50 maps, none of which uniformly resembled one another. The figure is not representative of all potential dataset maps, but this itself indicates a larger problem: Heat maps are isolated by example, and if this one were to be chosen in a random selection and used for explanation, almost no contribution could be made to model interpretability. Best viewed in color.*

model seeks and discriminates using the active eye region. When the region is occluded as in the second column, (displaying only lower facial data), the model uses information found around the mouth to determine classifications. The maps summarize where the DNN looks within an image in order to make predictions. While some generalizations can be made, the first column depicts a prominent issue with existing visualization techniques: inconsistency. Each sample produces a unique map, but due to lack of standardization, we can only draw imprecise conclusions about saliency from individual samples. Also, there is a risk of wide variation among sample constructions. The saliency map depicted in Figure 6.5 differs greatly from those shown in Figure 6.4 in terms of regions of interest. It also provides little specificity with regard to featural importance or information distribution.

6.4.5 Discussion: A Comparison with Human Perception

In section 2.2, we noted widely acknowledged decision characteristics of human gender classification. The majority of research has focused on organizing the contribution of various visual cues, with occlusion being the most popular method of evaluation. Human studies on cognition produce results which, by construction, can be directly compared to our own.

1. *Eyes as Predictors*: Nestor [90] and Russell [99] propose that the luminance difference between the main features - the eyes and mouth - and the rest of the face generates a pattern more typical of female than male faces. In other words, the greater the contrast between the luminance of the eyes and the other regions, the more likely a face will be considered female. The authors claim that the use of cosmetics is highly persuasive in this regard. As almost every candidate in CelebA darkens the eye region, this implies that human predictions of the eye region will be significantly female over male. This hypothesis is congruent with ResNet-50's observed class-skewing of eyes as female, which almost exactly follows the class distribution of genders in CelebA. Dupuis-Roy [100] applied the Bubbles technique to show that the eyes/eyebrow are the most important facial cue for accurate gender discrimination. So, to slightly oversimplify, we can assert that both the human and machine are more likely to identify eyes as male or female, and correctly classify gender on that basis

2. *Determinative Featural Importance*: Nestor et al.[91] attempts to use feature segmentation to diagnose the relative use of distinct local features, such as the eyes and mouth. Since we conclude that ResNet-50 prioritizes the learning of a specific facial region in horizontal passes, both machine and human processes use a few featurally important sections for gender recognition.

3. *Switching Attention*: Tanaka [68] experiments with facial inversions to suggest that the previously observed bias towards the eye region was attentional, and could be overridden by redirecting participant attention to the mouth. This almost identically parallels the heat maps in the previous section: when eye information is available, it is used deterministically for classification. However, when the eye/brows region is obscured, both the model and the human switch to reliance on the mouth for predictions.

4. *Common Occlusions*: Freud at al. [92, 101] test the effects of masks and sunglasses on human face processing, and remarks on the difference between lower and upper facial data availability. It is determined that due to lack of test subject exposure, humans recognize faces wearing masks much less accurately than those wearing sunglasses. Our analysis using permutation blackouts is versatile enough to form similar conclusions: ResNet-50 will recognize masked face *more* accurately than those wearing sunglasses. Our method gives explanations whose complexity can be scaled up or down depending on the need of the target audience, making them widely accessible.

In short, model perception is now entirely expressed in terms communicable to a human. Conclusively, our expression of model interpretability is not only able to provide generalizations explaining the behavior of ResNet-50, but can also form analogous responses to human face processing reactions.

6.5 Conclusion

We propose a novel framework for explaining the decisions of deep learning models targeting gender recognition. We use a methodological occlusion technique to construct machine explanations that closely resemble cataloged human decision

justifications. By converging on highly influential facial regions and extracting spatial information encoding, we show both the simplicity of our method and the informativeness of our results in comparison to existing works on interpretability. We demonstrate that our trends could be used to maximize model performance in cases of imperfect data. In future work, we will evaluate our metrics against a human baseline, and explore data augmentation strategies which leverage our findings

Chapter 7

DoppelVer: A Benchmark for Face Verification

7.1 Introduction

The task of face recognition has received considerable attention from computer vision and pattern recognition researchers in the past 20 years. This is because face identification has significant utility in the fields of biometrics, visual search, and socially assistive technologies [102, 103]. Additionally, compute equipment capable of running increasingly powerful algorithms has become relatively cheap and widely available. Face recognition technologies have significant impact on society with a market share of $5.69 billion worldwide in 2023 and a projected $12.05 billion by 2028 [104].

Work in face recognition and verification is dataset motivated. Every time a new dataset is released, there are significant improvements in face verification technology. Over the last several decades, there have been many datasets which have challenged the state-of-the-art (SOTA) face verification methods, such as

Labeled Faces in the Wild (LFW), IARPA Janus Benchmarks A, B, and C (IJB-{A,B,C}), etc. [105–108]. With the release of these datasets came a renewed interest in the field. Over the last few years, however, face identification on these datasets has reached a saturation point. For example, many methods achieve over 99% accuracy on the LFW benchmark. With such high accuracies we are able to visually inspect the samples that are incorrectly classified. In many cases these incorrectly classified samples are mislabeled meaning there is really no room for improvement on these datasets. In addition, face identification datasets are often collected with a focus on quantity, neglecting other important attributes. These problems provide the motivation for the proposed work.

This report introduces a new dataset – *DoppelVer* – consisting of unconstrained face images of doppelgangers – that is, individuals who look very similar and are often mistaken for each other. The purpose of DoppelVer is to challenge current SOTA facial feature extraction and face verification and identification methods. Although a plethora of datasets have been published to this end in the past decade, many of them are either unavailable or have been nearly solved. DoppelVer offers a specific challenge for modern face recognition methods, specifically the task of differentiating individuals who could pass for each other. To the best of our knowledge DoppelVer is the first dataset to increase face classification difficulty by increasing inter-class similarity rather than decreasing intra-class similarity. Upon publication of this paper, DoppelVer will be made publicly available.

Here we detail the highlights of the DoppelVer dataset, which will be expanded upon in the remainder of this work.

- DoppelVer contains 390 unique identities, each with at least one corresponding doppelganger pair.
- We provide the unaltered source images along with cropped, aligned, and centered (CCA) images.

- There is an average of 72 CCA samples per identity, with a minimum of 11 and a maximum of 98.
- For the CCA images we provide two evaluation protocols: doppelganger and **Vi**sual **S**imilarity from **E**mbeddings (ViSE). Under the doppelganger protocol negative samples are select images depicting an identity's doppelganger. The ViSE protocol uses a generalized image embedding model to select negative images that are highly visually similar to the current image sample.
- Both protocols are divided into 10 cross validation splits which are distinct across identities. The doppelganger protocol's cross validation splits are made up of 14,000 image pairs while ViSE's splits contain 3,500 image samples.

The remainder of the paper is organized as follows: in Section 7.2 we provide background to the field of face recognition, with a focus on feature extraction and face classification methods. Section 7.2 also details similar datasets and the novelty of DoppelVer. Section 7.3 contains a more detailed description of the DoppelVer dataset including data collection, pre-processing, labeling, and the generation of the evaluation protocols. In Section 7.4 we provide results of our experimentation comparing the performance of SOTA facial recognition pipelines on existing benchmark datasets and DoppelVer.

7.2 Related Work

7.2.1 Background

Face recognition is separated into three well-defined steps: (1) face detection and localization, (2) extraction of features from the detected face, and (3) classification

(verification or identification) [103]. The first task is to decide whether or not there are faces in an image. If there are one or more faces, then the system identifies bounding boxes for each face. The feature extraction step generates a feature vector from the localized face. This feature vector should be discriminative enough to separate images of one identity from images of other identities. Lastly, there is the classification step. This is separated into two classes of techniques: identification and verification. In the identification scenario the system is aware of a finite number of identities and it should learn to match each image sample to one identity class. For the verification task the model is only provided with supervision in the form of a binary label which represents either same or different, and so pairs of images are compared at each step.

Any face recognition system that is meant to be deployed in "the wild" will need to perform all three of these steps. That being said, each step is commonly considered an active research topic. The intended purpose of the DoppelVer dataset is to contribute towards improvements in the final two steps. In this work, we devote our efforts towards the feature extraction and classification tasks. This is because most modern methods employ deep learning techniques, which combine feature extraction and classification into a single system. Additionally, research has seemingly slowed in these areas.

One might suggest that the field of face classification is reaching its maturity, citing results on the well-known benchmarks such as LFW, AgeDB, or IJB-{A,B,C} [105–109]. Rather than assuming that the reported metrics are due to the techniques solving the task of visually recognizing faces, we hypothesize that the modern techniques have improved beyond the level of difficulty provided by the current benchmarks. For example, in 2015 Liu et al. published a result of 99.77% accuracy on the LFW benchmark [110]. The dataset's evaluation protocol contains only 6000 images. This means that for nearly a decade methods have been attempting

to show improvements on a method that mis-classifies only 14 images, five of which are known to be incorrectly labeled.

Other methods have emerged with the intent of contributing to the issue of increasing unconstrained face recognition benchmark difficulty [109, 111–113]. These methods primarily focus on increasing difficulty of the classification task with highly varied pose and age. These features essentially decrease the intra-class similarity (i.e. selecting images of the same identity that are visually different).

Our DoppelVer dataset increases classification difficulty by increasing inter-class similarity (i.e. selecting images of different identities that are visually similar). We accomplish this goal in two distinct ways. First, we aggregate doppelganger pairs. A doppelganger pair is simply two individuals who have similar facial features. This protocol is constructed by human labelers selecting visually similar identities. Second, for a given image we mine a negative sample which is highly visually similar. This is accomplished by generating an embedding or latent vector for all images in the dataset. We search for pairs of images whose embeddings are near one another in the latent space. By these two methods we produce two protocols that we have named doppelganger and **Vi**sual **S**imilarity from **E**mbeddings (ViSE).

7.2.2 Existing Datasets

There are a large number of datasets collected and presented for the purpose of facial feature extraction and classification. Many of these datasets are designed either for training or evaluation. Here we describe the major datasets that already exist for the purpose of model evaluation and benchmarking and compare them with the proposed DoppelVer dataset.

Labeled Faces in the Wild (LFW) [105]: The LFW dataset was created by Huang et al. in 2007. At the time of publishing, many face recognition datasets

were collected by small teams of researchers with the intent of collecting facial images in constrained settings. LFW however was meant for studying the problem of recognizing faces in unconstrained settings. The dataset contains 13,233 images and 7,549 identities. The researchers behind LFW contributed significantly to the field by presenting a dataset organization that focused on the honest reporting of results for the task of open-set face recognition. Their dataset contains a development view and an evaluation view as well as splits for 10 fold cross-validation. The current SOTA accuracy on LFW is 99.8% (±0.2001) [114].

AgeDB [109]: This dataset was introduced in 2017, with a focus on accurate hand-labeling of age. This is a useful database when performing tasks such as age-invariant face verification, age estimation, and face age progression. The database contains 16,488 images of 568 identities with accurate-to-the-year age labels. The average number of images per individual is 29, with an age range of 1 to 101 years old, the average age for an individual being 50.3 years. AgeDB provides four face verification protocols, each split into 10 folds following LFW's process. These four protocols restrict the age variance across sample pairs. The provided protocols cap age range to 5, 10, 20 and 30 years respectively. The current SOTA accuracy on AgeDB 30 is 98.7% [115]

Cross-Age LFW (CA-LFW) [112]: The authors of this database posit that methods reporting accuracy on LFW's benchmark are optimistic. To show this, CA-LFW has both positive and negative pairs which depict a large age gap, while also providing negative pairs which are of the same race and gender. These visually similar negative pairs emphasize the effect of age difference on classifier performance. This dataset contains the same identifies as LFW with 6,000 image pairs. The current SOTA accuracy on CA-LFW is 95.87% [116]

Cross-Pose LFW (CP-LFW) [113]: CP-LFW was proposed by the same authors as CA-LFW and was released one year later. This publication shifts focus to

the important task of face verification in the presence of extreme pose. They note that nearly all images in LFW are near-frontal, suggesting that results on LFW provide a poor representation of a face recognition method's performance when deployed into a real setting. The current SOTA accuracy on CA-LFW is 92.08% [116]

Each of the databases detailed above provide an important contribution to furthering the field of face recognition. These datasets provide unconstrained images and in the cases of [109, 111–113] the sample pairs vary along specific axis which were not well represented in LFW. As mentioned previously, these datasets focus on selecting positive pairs which are visually dissimilar to one another. DoppelVer's goal is to expand on a dimension of challenge which has not yet been addressed. This dimension is that of visual similarity among negative samples. This yet unseen challenge will force methods to extract significantly more fine-grained, prominent features from face images. In order to achieve high performance on DoppelVer, techniques will be required to extract those features which uniquely define a given identity.

7.3 Proposed Method

7.3.1 Dataset Collection

In order to construct a dataset for which negative samples are analogous to positive samples it is intuitive to begin by aggregating a list of identities which bare visual similarity to human labelers (i.e. doppelgangers). Doppelganger identity pairs were collected through labeler intuition of similar looking identities and lists of doppelgangers publicly available on the Internet. We present a large list of doppelganger identity combinations, totalling 237 pairs and 390 individuals. For

each individual, 100 images were scraped from online sources. The average number of images presented in the dataset for each person is approximately 72 due to pruning of noisy samples and duplicates.

7.3.2 Data Preparation

Data preparation involved two distinct steps: (1) cropping, aligning and centering the images, and (2) hand removal of erroneous samples and duplicate images.

7.3.2.1 Cropping, aligning and centering:

The first step in the data preparation is to reduce the original images into cropped, aligned, and centered images. We crop to remove information which is extraneous to the face recognition task. Alignment and centering are performed as they have been recognized as important for achieving competitive face recognition benchmark performance. Alignment involves rotating the image such that the eyes lie on a horizontal line (i.e. the same y-coordinates). The operation of centering moves the face in the frame of the image such that it appears centrally. Centering is accomplished by repeating edge pixels along either the horizontal or vertical borders of the image. The cropping operation relies on a bounding box and centering/alignment rely on facial landmarks. We extract the bounding boxes and facial landmarks for images in DoppelVer with the MTCNN detector [117].

While processing the dataset with MTCNN, three cases may occur: (1) MTCNN does not detect a face, (2) MTCNN detects a single face, and (3) MTCNN detects multiple faces. Images where a face is not detected are pruned from the dataset. Although MTCNN returns a detected face in most images, not all detections contain the target identity or a valid face. Each detection is hand-checked for validity during the cleaning phase of pre-processing. When at least one face is

detected, MTCNN returns a bounding box for the image along with five facial landmarks. The landmarks provide the detected location of the centers of the eyes, corners of the mouth, and tip of the nose.

Initially we cropped the source images to the bounding boxes predicted by MTCNN, but found that the crop was too tight. These crops often removed valuable information such as the top of head, ears and most of the neck. We expand MTCNN's detected bounding box width and height by 50%. This produces crops which contain more contextual information. There are cases for which the detected face is near the border of the image, restricting our ability to expand the bounding box. In these cases we simply set the desired bounding box location to the border of the image.

After cropping, we align the images according to the extracted landmark locations. Our alignment rotates the images such that the detected landmark for left and right eyes have the same y-axis coordinate. During the alignment process some image information is lost due to the corners of the image rotating outside of the frame. Following the lead of the CelebA dataset, we reduce the effects of this information loss by performing same padding for any pixels that are lost due to rotation [1].

The last pre-processing step is to center the image so that the center most pixel of the image is within the bounds of the detected face. Centering is performed by computing a landmark which lies at the mid-point between the left and right eye landmarks. Additional pixels are appended to the horizontal and vertical image borders such that the center of the face is equidistant to each border. The appended pixels are simply duplicates of the pixels which are along the border that needs to be expanded.

7.3.2.2 Removal of erroneous or duplicate Images:

We remove unsatisfactory images by hand and by automatic detection. In the case of hand labeling, labelers began with the original image set collected from the internet. Their task was to pass over the images and delete any image which contained erroneous detections (e.g. not depicting the correct identity or images not containing a face). The set of images which had complete labeler agreement was accepted. The set of images which did not have agreement were re-labeled. Any remaining images which the labelers did not reach agreement on were pruned from the dataset. The images which achieved hand label agreement were passed to the automatic detection system.

The automatic detection system works by generating embeddings for each face image in the dataset with the dinov2s model [118]. dinov2s is a general purpose image embedding model, built to capture a discriminative representation of input images without finetuning. The cosine similarity is computed between all combinations of input images' embeddings to determine samples which are highly visually similar. To compute the embeddings and cosine similarities efficiently we utilize the fastdup library [119] from Visual Layer. For any image pair that has exact similarity (i.e. duplicate images), one image from the pair is pruned from the dataset. Next, we return all of the image pairs that are above a threshold of 0.92 similarity. We extract these images pairs and provide them to human labelers to find near duplicate images (i.e. images that have been horizontally flipped, color jittered, cropped slightly differently, etc.), which are removed from the dataset.

7.3.3 Protocol Generation

The DoppelVer dataset contains in total 27,967 carefully curated and processed images. The question that remains is the best way to utilize these images for

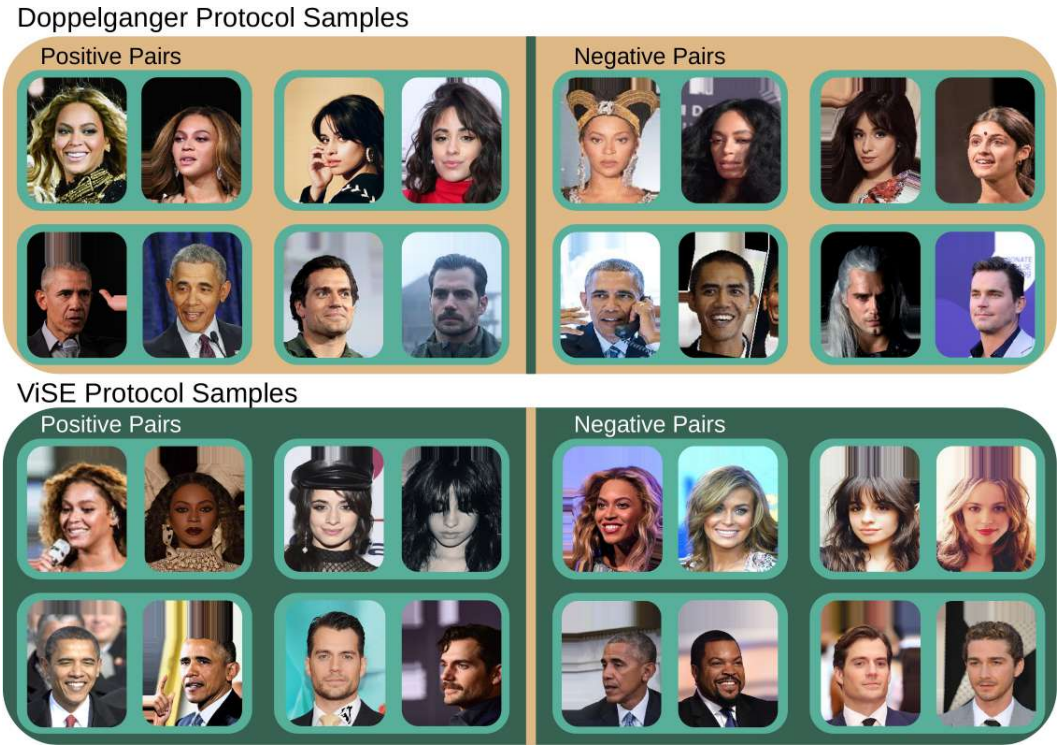


FIGURE 7.1: Shown above are samples from both protocols of the DoppelVer dataset – doppelganger and ViSE. We note that negative samples from the Doppelganger protocol share facial attributes while the image pairs in ViSE frequently share factors external to the face such as pose, clothing, and background.

assessing and benchmarking feature extraction and face classification methods. To answer this question, we introduce two protocols for evaluation using DoppelVer: doppelganger and ViSE. Fig. 7.1 provides example image pairs for each protocol in DoppelVer and Fig. 7.2 shows samples from CA-LFW and CP-LFW.

Both protocols are made up of positive and negative image pairs. Positive image pairs in both protocols signify instances where both images depict the same identity. In the doppelganger protocol, negative pairs are made up of one image sample depicting the current target identity and one image sample depicting their doppelganger. In the ViSE protocol the negative pairs contain an image sample depicting an identity which does not generally appear as visually similar to the current identity, but in a one-off case is visually similar. Such similarity often arises due to comparable pose, lighting, hair style, clothing, or image background. After generating a large number of image pairs, we divide the dataset into 10

FIGURE 7.2: The upper portion of this figure presents samples from the CA-LFW dataset and the lower portion contains samples from CP-LFW. The CA-LFW samples showcase differences in age while CP-LFW's images showcase differences in pose.

equally sized splits. Each split is divided such that images of an identity are in only a single split. Identities are divided the same in each protocol (e.g. split 0 of the doppelganger protocol depicts the same identities as split 0 of ViSE).

The doppelganger protocol is generated with our curated list of doppelganger pairs. We create the pair instances in the doppelganger protocol as follows. First, we sample 500 image combinations, without replacement, for every pair of doppelgangers and identities with themselves. After generating all pairs following this criteria we separate the samples into 10 splits based on their identities and pairs such that the same identity never shows up in multiple splits. Approximately 10 percent of the dataset is placed into each split. Finally, from each split we randomly sample 7,000 positive pairs and 7,000 negative pairs. We do this to follow the procedures laid out by LFW. This protocol has a positive label and negative label ratio of exactly 50%. It has a gender distribution of 44.96% males and 55.04% female samples respectively. Identities in each split have a relatively even representation with an average minimum contribution of 4.31%, average maximum

contribution of 19.07%, and an average standard deviation between representation of 5.32%. In total the doppelganger protocol has 140,000 sample pairs.

To generate the ViSE protocol we use a similar approach to the one described in the automatic detection of unsatisfactory images. We begin by generating embeddings for each image in the dataset with the dinov2s model. Next, we compute the cosine similarity between images which do not come from the same identity. We retain all image pairs that have a similarity greater than 0.80. We have found that this form of mining hard pairs image by image rather than individual by individual results in significantly more visual similarity between image pairs. Using the same identities in each split as the doppelganger protocol, we break the protocol into 10 splits with unique identities in each split. This protocol has a positive label and negative label ratio of exactly 50%. It has a gender distribution of 40.36% male and 59.64% female. Identities in each split have a relatively even representation with an average minimum contribution of 2.29%, average maximum contribution of 17.61%, and an average standard deviation between representation of 3.6%. This protocol has 35,000 verification pairs.

7.3.4 Intended Use

The DoppelVer dataset is intended to provide a new challenge for the research community developing methods in the area of facial recognition. DoppelVer has been designed to act as an evaluation dataset, not a training dataset. In the past decade the most effective methods of facial recognition have utilized large training sets such as CASIA-WebFace, MegaFace, VGGFace2, MS-Celeb-1M [120–123]. These datasets contain 34.94K, 1.03M, 3.31M, 10M samples respectively. Although an aggregate of visually difficult pairs is attractive for faster convergence time, DoppelVer does not contain enough diversity to effectively and ethically train models.

We provide cross validation splits for both protocols in DoppelVer. The purpose of these splits is two-fold. First, some methods may wish to perform feature extraction prior to face classification. Such extraction methods should pre-train on external sources and infer features for each image in DoppelVer. At evaluation time final-stage classifiers should be iteratively trained from scratch (using their pre-trained feature extraction methods) on nine splits and evaluated on the tenth. Performance should be recorded as an average across the 9 models. We refer to interaction with the dataset in this way as **View 1**. Second, methods that wish to train on external data and perform only evaluation on DoppelVer should use split 0 for algorithm development and validation of results. The model should not be exposed to data in any of the other nine splits until final evaluation. Use of the dataset in this way is called **View 2**.

Taking motivation from the LFW dataset, we suggest that researchers utilizing **View 1** report estimated mean accuracy (EM ACC) and standard error of the mean (SEM). We define these metrics in the following way:

$$\hat{\mu} = \frac{\Sigma_{i=1}^{9} p_i}{9}, SEM = \frac{\hat{\sigma}}{\sqrt{9}}, \hat{\sigma} = \sqrt{\frac{\Sigma_{i=1}^{9}(p_i - \hat{\mu})^2}{9}}$$

where p_i is the percentage of correct classifications on **View 1** when using the i^{th} split for testing. $\hat{\sigma}$ is the estimate of the standard deviation. As noted by the authors of LFW, it is important that accuracy is computed with parameters and thresholds chosen independently of the test data. Researchers should not simply choose the point on a Precision-Recall curve giving the highest accuracy.

For the methods which utilize **View 2** of DoppelVer, we advocate for the use of accuracy (ACC) and area under the receiver operating characteristic curve (ROC AUC). We elect for the use of ACC and ROC AUC because of the balanced nature of classes in the Doppelganger and ViSE protocols. In addition, the correct classification of true positives is equally important to classification of true negatives.

7.4 Experiments

In this section, we highlight the challenges posed by the DoppelVer dataset as compared to other existing evaluation datasets. We detail the methods used for evaluation, the training data, and the process employed for training and testing.

7.4.1 Evaluation Model

To provide an accurate depiction of the challenge posed by DoppelVer, it is important that we evaluate DoppelVer with SOTA face recognition models. Due to ease of implementation and competitive results we have elected to utilize the techniques described by Wen et al. in SphereFace2 [124]. In particular we train the 20 layer SphereFace Network (SFNet-20), initially proposed in [125], with the following loss functions: COCO, SphereFace, CosFace, ArcFace, and SphereFace2. Following Wen et al., we equip SFNet-20 with batch normalization to facilitate model optimization. A complete implementation for training SFNet-20 with the aforementioned loss functions can be found in the OpenSphere GitHub repository [126].

7.4.2 Training and Evaluation Process

For pre-processing, we crop face images in each dataset with MTCNN, resize images to a size of 112×112, and normalize each RGB pixel [0, 255] to the range [-1, 1]. We trained our models on a single Nvidia Geforce RTX 3090 GPU. Each model is trained for 70,000 batches of size 512. The model weights are updated by stochastic gradient descent with a momentum of 0.9 and weight decay of 0.0005. The initial learning rate of 0.1 is reduced by a factor of 0.1 at batches 40,000; 60,000; and 70,000.

We evaluate our dataset and protocols with VGGFace2, MS-Celeb-1M, and CASIA-WebFace [120, 122, 123]. In each run the VGGFace2 dataset was found to produce the best results on each evaluation dataset. VGGFace2 contains between 80 and 800 images for each identity making it a powerful training dataset for the face verification task. Evaluation of the trained models is performed on LFW, CA-LFW, CP-LFW, AgeDB 30, view 2 of DoppelVer's doppelganger protocol, and view 2 of DoppelVer's ViSE protocol. Our measured accuracy and ROC AUC are provided in Tables 7.1 and 7.2 respectively.

7.4.3 Discussion of Results

We are satisfied with the performance achieved by the SOTA methods on the existing benchmark datasets. SOTA performance on the LFW dataset is 99.8% accuracy. Our training of SphereFace achieves an accuracy of 99.58%, mis-classifying just 25 samples. With this result we can be assured that this baseline is competitive

TABLE 7.1: Average accuracy of face verification for the comparison models trained with VGGFace2 and benchmarked on various datasets.

Method	LFW	CA-LFW	CP-LFW	AgeDB	Doppelganger	ViSE
COCO [127]	99.08	91.25	88.48	89.40	61.14	52.53
SphereFace [125]	99.58	93.15	91.65	93.53	63.48	57.08
CosFace [128]	99.52	93.03	91.37	93.02	63.29	56.93
ArcFace [129]	99.55	93.40	91.18	92.57	63.28	57.70
SphereFace2 [124]	99.53	93.80	90.83	93.38	61.66	55.41
Average	**99.45**	**92.93**	**90.70**	**92.38**	**62.57**	**55.93**

TABLE 7.2: Average AUC of face verification for the comparison models trained with VGGFace2 and benchmarked on various datasets.

Method	LFW	CA-LFW	CP-LFW	AgeDB	Doppelganger	ViSE
COCO [127]	99.89	96.56	93.57	96.03	65.13	50.53
SphereFace [125]	99.92	97.44	95.50	98.11	68.65	59.41
CosFace [128]	99.91	97.28	95.64	97.86	67.91	58.58
ArcFace [129]	99.89	96.99	95.46	97.53	68.15	59.79
SphereFace2 [124]	99.89	97.55	95.42	98.02	65.43	55.77
Average	**99.90**	**97.16**	**95.12**	**97.51**	**67.05**	**56.82**

with other SOTA methods. The best published results on the other benchmark datasets are 95.87%, 92.08%, and 98.7% accuracy on CA-LFW, CP-LFW, and AgeDB 30 respectively. Regardless of loss function, the baseline networks struggle significantly more with variations in pose than variations in age. CA-LFW and AgeDB appear to present a similar degree of difficulty to the models.

It is clear from our experiments that the doppelganger and ViSE protocols of DoppelVer are much more difficult for the classifiers than the other datasets. Results are better for the doppelganger protocol than the ViSE protocol. This result aligns with intuition. Two identities that are doppelgangers may in general share facial attributes, but variations in clothing, hair style, lighting, and facial expression are expected when viewing a gallery of images depicting them.

On the other hand, the ViSE protocol contains image pairs which are adversarial in nature. By this we mean that the combinations of samples are those which a deep network is expected to struggle to differentiate. Although we use a different deep convolutional network to select samples which are visually similar than we do for performing facial recognition, one would expect that the visual features which are attended to by deep networks would have some similarity.

We believe that methods which will perform well on the ViSE protocol will need to extract features which are highly specific to the task of facial recognition. In addition, methods will need to not only detect relevant facial features, but discern if the features are prominent/defining to the individual's face.

7.5 Conclusion

In this work we introduce DoppelVer, a novel evaluation dataset for the tasks of facial feature extraction and face verification. DoppelVer consists of 27,967 carefully curated face images, which are used in two face verification protocols

of image pairs: doppelganger and ViSE. We evaluate our methods using several SOTA methods. A near SOTA baseline model is only capable of correctly performing face verification at an accuracy of 62.57% and 55.93% in the doppelganger and ViSE protocols respectively. This indicates that despite impressive results on popular benchmark datasets, there is still work to be done in the field of facial recognition.

Future research should explore improvements to deep vision models to enable accurate classification of visually similar individuals. Additionally, future work might involve the application of the ViSE protocol's adversarial image pair selection to larger selections of facial data to enable the training of deep networks with visually similar negative pairs. Lastly, this data might be used to understand the difference in vision model perceptions between images of identical twins or parents and children at similar times of life.

Chapter 8

Studying the Representations of Facial Recognition Models in Visually Similar Environments

Chapter 9

Conclusion

Conclusion....

Chapter 10

Future Research

Future work.....

Bibliography

[1] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

[2] E. Hand, C. Castillo, and R. Chellappa, "Doing the best we can with what we have: Multi-label balancing with selective learning for attribute prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, April 2018.

[3] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020.

[4] N. Kumar, P. Belhumeur, and S. Nayar, "Facetracer: A search engine for large collections of images with faces," in *European Conference on Computer Vision*, pp. 340–353, Springer, 2008.

[5] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar, "Attribute and simile classifiers for face verification," in *International Conference on Computer Vision*, pp. 365–372, IEEE, 2009.

[6] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable visual attributes for face verification and image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1962–1977, 2011.

[7] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[8] E. M. Rudd, M. Günther, and T. E. Boult, "Moon: A mixed objective optimization network for the recognition of facial attributes," in *European Conference on Computer Vision*, pp. 19–35, Springer, 2016.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[10] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *CoRR*, vol. abs/1505.04366, 2015.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[12] M. M. Kalayeh, B. Gong, and M. Shah, "Improving facial attribute prediction using semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 4227–4235, IEEE, 2017.

[13] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2021.

[14] Y.-H. Tseng and S.-S. Jan, "Combination of computer vision detection and segmentation for autonomous driving," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 1047–1052, 2018.

[15] G. Brazil, X. Yin, and X. Liu, "Illuminating pedestrians via simultaneous detection and segmentation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4960–4969, 2017.

[16] F. Flohr, D. Gavrila, *et al.*, "Pedcut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues.," in *BMVC*, 2013.

[17] X. Zhu, H.-I. Suk, and D. Shen, "A novel matrix-similarity based loss function for joint regression and classification in ad diagnosis," *NeuroImage*, vol. 100, pp. 91–105, 2014.

[18] X. Zhu, H.-I. Suk, S.-W. Lee, and D. Shen, "Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 607–618, 2016.

[19] Y. Gao, Y. Shao, J. Lian, A. Z. Wang, R. C. Chen, and D. Shen, "Accurate segmentation of ct male pelvic organs via regression-based deformable models and multi-task random forests," *IEEE Transactions on Medical Imaging*, vol. 35, no. 6, pp. 1532–1543, 2016.

[20] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests.," in *BMVC*, pp. 1–10, 2008.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing*

and Computer-Assisted Intervention – MICCAI 2015 (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), (Cham), pp. 234–241, Springer International Publishing, 2015.

[23] J. Warrell and S. Prince, "Labelfaces: Parsing facial features by multiclass labeling with an epitome prior," pp. 2481–2484, 11 2009.

[24] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller, "Augmenting crfs with boltzmann machine shape priors for image labeling," 06 2013.

[25] B. Smith, l. Zhang, J. Brandt, Z. Lin, and J. Yang, "Exemplar-based face parsing," pp. 3484–3491, 06 2013.

[26] Y. Liu, H. Shi, Y. Si, H. Shen, X. Wang, and T. Mei, "A high-efficiency framework for constructing large-scale face parsing benchmark," *CoRR*, vol. abs/1905.04830, 2019.

[27] P. Luo, X. Wang, and X. Tang, "Hierarchical face parsing via deep learning," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2480–2487, 2012.

[28] B. M. Smith, L. Zhang, J. Brandt, Z. Lin, and J. Yang, "Exemplar-based face parsing," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3484–3491, 2013.

[29] J. Lin, H. Yang, D. Chen, M. Zeng, F. Wen, and L. Yuan, "Face parsing with roi tanh-warping," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 5647–5656, IEEE Computer Society, jun 2019.

[30] S. Liu, J. Shi, J. Liang, and M. Yang, "Face parsing via recurrent propagation," *CoRR*, vol. abs/1708.01936, 2017.

[31] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005.

[33] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[34] A. Zadeh, T. Baltrusaitis, and L. Morency, "Deep constrained local models for facial landmark detection," *CoRR*, vol. abs/1611.08657, 2016.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[37] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," 1967.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[39] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite mixture models," *Annual Review of Statistics and Its Application*, vol. 6, no. 1, pp. 355–378, 2019.

[40] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, (Cambridge, MA, USA), p. 849–856, MIT Press, 2001.

[41] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, pp. 504 – 507, 2006.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[43] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *international conference on machine learning*, pp. 3861–3870, PMLR, 2017.

[44] M. M. Fard, T. Thonet, and E. Gaussier, "Deep k-means: Jointly clustering with k-means and learning representations," *Pattern Recognition Letters*, vol. 138, pp. 185–192, 2020.

[45] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," *CoRR*, vol. abs/1511.06335, 2015.

[46] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1753–1759, 2017.

[47] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *Neural Information Processing* (D. Liu, S. Xie, Y. Li,

D. Zhao, and E.-S. M. El-Alfy, eds.), (Cham), pp. 373–382, Springer International Publishing, 2017.

[48] W. Wang, D. Yang, F. Chen, Y. Pang, S. Huang, and Y. Ge, "Clustering with orthogonal autoencoder," *IEEE Access*, vol. 7, pp. 62421–62432, 2019.

[49] S. Affeldt, L. Labiod, and M. Nadif, "Spectral clustering via ensemble deep autoencoder learning (sc-edae)," *Pattern Recognition*, vol. 108, p. 107522, 2020.

[50] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," *CoRR*, vol. abs/1807.05520, 2018.

[51] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognition*, vol. 83, pp. 161–173, 2018.

[52] D. Gupta, R. Ramjee, N. Kwatra, and M. Sivathanu, "Unsupervised clustering using pseudo-semi-supervised learning," in *International Conference on Learning Representations*, 2020.

[53] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *CoRR*, vol. abs/1611.02648, 2016.

[54] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang, "Deep adversarial multi-view clustering network," pp. 2952–2958, 08 2019.

[55] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu, "Subspace learning for unsupervised feature selection via matrix factorization," *Pattern Recognition*, vol. 48, no. 1, pp. 10–19, 2015.

[56] F. Saberi-Movahed, M. Eftekhari, and M. Mohtashami, "Supervised feature selection by constituting a basis for the original space of features and matrix factorization," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 7, pp. 1405–1421, 2020.

[57] H. Steck, "Autoencoders that don't overfit towards the identity," in *NeurIPS*, 2020.

[58] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst*, vol. 14, 04 2002.

[59] S. Wan, J. Kim, and K. Won, "Sharp: hyper-fast and accurate processing of single-cell rna-seq data via ensemble random projection," *Genome Research*, vol. 30, p. gr.254557.119, 01 2020.

[60] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering: methods and analysis," *Knowledge and Information Systems*, vol. 51, pp. 661–689, 9 2017.

[61] K. Tian, S. Zhou, and J. Guan, "Deepcluster: A general clustering framework based on deep learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 809–825, Springer, 2017.

[62] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[63] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[64] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[65] J. Cheng, P. Wang, G. Li, Q. Hu, and H. Lu, "Recent advances in efficient computation of deep convolutional neural networks," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, pp. 64–77, 2018.

[66] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[67] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, 2019.

[68] J. Tanaka, M. Kaiser, S. Hagen, and L. Pierce, "Losing face: Impaired discrimination of featural and configural information in the mouth region of an inverted face," *Attention, perception psychophysics*, vol. 76, 01 2014.

[69] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[70] I. Pointer, "Class activation mappings in pytorch."

[71] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, 2018.

[72] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke,

Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[74] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, pp. 31 – 57, 2018.

[75] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," *ACM Computing Surveys (CSUR)*, vol. 51, pp. 1 – 42, 2019.

[76] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *European Conference on Computer Vision(ECCV)*, vol. 8689, pp. 818–833, 01 2013.

[77] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 12 2014.

[78] A. Mahendran and A. Vedaldi, "Salient deconvolutional networks," vol. 9910, pp. 120–135, 10 2016.

[79] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, pp. 336–359, 2019.

[80] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.

[81] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? - weakly-supervised learning with convolutional neural networks," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 685–694, 2015.

[82] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3449–3457, 2017.

[83] M. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," pp. 97–101, 02 2016.

[84] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 4768–4777, Curran Associates Inc., 2017.

[85] H. S. J. E. S. S. Slack, Dylan and H. Lakkaraju, "Fooling lime and shap: Adversarial attacks on post hoc explanation methods," 02 2020.

[86] N. Thom and E. Hand, *Facial Attribute Recognition: A Survey*, pp. 447–459. 10 2021.

[87] B. Golomb, D. Lawrence, and T. Sejnowski, "Sexnet: A neural network identifies sex from human faces.," pp. 572–579, 01 1990.

[88] G. Cottrell and J. Metcalfe, "Empath: Face, emotion, and gender recognition using holons," in *NIPS*, 1990.

[89] C.-B. Ng, Y. H. Tay, and B.-M. Goi, "Recognizing human gender in computer vision: A survey," vol. 7458, pp. 335–346, 09 2012.

[90] A. Nestor and M. J. Tarr, "Gender recognition of human faces using color," *Psychological Science*, vol. 19, pp. 1242 – 1246, 2008.

[91] A. Nestor and M. Tarr, "The segmental structure of faces and its use in gender recognition," *Journal of vision*, vol. 8, pp. 7.1–12, 02 2008.

[92] E. Freud, A. Stajduhar, R. S. Rosenbaum, G. Avidan, and T. Ganel, "The covid-19 pandemic masks the way people perceive faces," *Scientific Reports*, vol. 10, 2020.

[93] D. Maurer, R. Le Grand, and C. Mondloch, "The many faces of configural processing," *Trends in cognitive sciences*, vol. 6, pp. 255–260, 07 2002.

[94] R. Wang, J. Li, H. Fang, M. Tian, and J. Liu, "Individual differences in holistic processing predict face recognition ability," *Psychological science*, vol. 23, pp. 169–77, 02 2012.

[95] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12 2014.

[96] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[97] L. Biewald, "Experiment tracking with weights and biases," 2020. Software available from wandb.com.

[98] k. keyurr2, "Facial-landmarks: Facial landmarks detection with opencv, dlib, dnn."

[99] R. Russell, "Sex, beauty, and the relative luminance of facial features," *Perception*, vol. 32, pp. 1093–107, 02 2003.

[100] N. Dupuis-Roy, I. Fortin, D. Fiset, and F. Gosselin, "Uncovering gender discrimination cues in a realistic setting.," *Journal of vision*, vol. 9 2, pp. 10.1–8, 2009.

[101] E. Noyes, J. Davis, N. Petrov, K. Gray, and K. Ritchie, "The effect of face masks and sunglasses on identity and expression recognition with super-recognizers and typical observers," *Royal Society Open Science*, vol. 8, 03 2021.

[102] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics*, vol. 9, no. 8, 2020.

[103] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2, 2020.

[104] MordorIntelligence, July 2023.

[105] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[106] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1931–1939, 2015.

[107] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother, "Iarpa janus benchmark-b face dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 592–600, 2017.

[108] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother, "Iarpa janus benchmark - c: Face dataset and protocol," in *2018 International Conference on Biometrics (ICB)*, pp. 158–165, 2018.

[109] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "Agedb: the first manually collected, in-the-wild age database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, vol. 2, p. 5, 2017.

[110] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting Ultimate Accuracy: Face Recognition via Deep Embedding," *arXiv e-prints*, p. arXiv:1506.07310, June 2015.

[111] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9, 2016.

[112] T. Zheng, W. Deng, and J. Hu, "Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments," *CoRR*, vol. abs/1708.08197, 2017.

[113] T. Zheng and W. Deng, "Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments," Tech. Rep. 18-01, Beijing University of Posts and Telecommunications, February 2018.

[114] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. Zweiri, and N. Werghi, "Ghostfacenets: Lightweight face recognition model from cheap operations," *IEEE Access*, vol. 11, pp. 35429–35446, 2023.

[115] X. An, X. Zhu, Y. Gao, Y. Xiao, Y. Zhao, Z. Feng, L. Wu, B. Qin, M. Zhang, D. Zhang, *et al.*, "Partial fc: Training 10 million identities on a single machine," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1445–1449, 2021.

[116] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

[117] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[118] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," *arXiv e-prints*, p. arXiv:2304.07193, Apr. 2023.

[119] visual layer, "fastdup." https://github.com/visual-layer/fastdup, July 2023.

[120] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014.

[121] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4873–4882, 2016.

[122] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pp. 67–74, 2018.

[123] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 87–102, Springer International Publishing, 2016.

[124] Y. Wen, W. Liu, A. Weller, B. Raj, and R. Singh, "Sphereface2: Binary classification is all you need for deep face recognition," *CoRR*, vol. abs/2108.01513, 2021.

[125] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6738–6746, 2017.

[126] ydwen, "Opensphere." https://github.com/ydwen/opensphere, July 2023.

[127] Y. Liu, H. Li, and X. Wang, "Rethinking feature discrimination and polymerization for large-scale recognition," *CoRR*, vol. abs/1710.00870, 2017.

[128] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018.

[129] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 5962–5979, 2022.