Open-Set Face Recognition with Entropic Open-Set Loss

Master's Thesis

Yu Linghu

19-771-682

Submitted on
February 1, 2022

Thesis Supervisor
Prof. Dr. Manuel Günther

Artificial Intelligence MACHINE Learning

University of Zurich^UZH

Department of Informatics

Master's Thesis

Author: Yu Linghu, yu.linghu@uzh.ch

Project period: August 9, 2021 - February 1, 2022

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

I thank Prof. Dr. Manuel Günther's supervision on the entire thesis. Thank for his time to our bi-weekly discussion and his detailed, but fast response and explanation for my questions. I thank Dr. Tiago de Freitas Pereira and IDIAP Research Institute and their good infrastructure Bob, which half of my experiments rely on. I thank my friend, Xinyi Zhang, for her support in my two-year Master's study. Thank for my family's support and sibling's advice so that I chose the University of Zurich for the past two year's life.

Abstract

The goal for the open-set face recognition is to identify the unseen subjects and do not assign them to any known subject with high confidence. There are two types of subjects involved in the task: the ones that we are interested in and have labels, i.e. known subjects; the ones that we do not care about and have no labels (we use -1 in the experiment instead), i.e. unknown subjects. We build a complete face recognition pipeline through Bob. ArcFace R100 network, as a feature extractor, has a good performance on the IJB-C dataset. Our goal is to add an extra network after ArcFace to enhance its power on open-set face recognition tasks. We attempt three cases: first, the unknown subjects have never appeared in the training; second, the unknown subjects appear in both training and testing; third, the unknown subjects only appear in the testing, and they are replaced by the adversarial samples generated from the knowns in the training. The training unknowns have no overlap with the testing unknowns in case three. Plain softmax loss and entropic open-set loss are applied to the first two cases, respectively, and objectosphere loss is used for the second and third cases. We prove that those models create a high True Positive Identification Rate especially when the False Positive Identification Rate is small. Replacing the unknown subjects in case two to the adversarial samples as in case three is successful without performance degradation. One flaw is that the magnitude separation property of the entropic open-set loss and objectosphere loss is not apparent. When working with the adversarial samples, the situation is worse.

Zusammenfassung

Das Ziel der open-set Gesichtserkennung ist es, die ungesehenen Personen zu identifizieren und sie keiner bekannten Person mit hoher Konfidenz zuzuordnen. Es gibt zwei Arten von Personen, die an der Aufgabe beteiligt sind: diejenigen, an denen wir interessiert sind und die Labels haben, d.h. die bekannte Personen; diejenigen, die uns nicht interessieren und keine Labels haben (wir verwenden stattdessen -1 im Experiment), d.h. die unbekannte Personen. Wir bauen eine komplette Gesichtserkennungspipeline durch Bob auf. Das ArcFace R100 Netzwerk zeigt als Merkmalsextraktor eine gute Leistung an dem IJB-C Datensatz. Unser Ziel ist es, ein zusätzliches Netzwerk nach ArcFace hinzuzufügen, um seine Leistung bei Aufgaben der open-set Gesichtserkennung zu verbessern. Wir versuchen drei Fälle: Erstens, die unbekannten Personen sind nie im Training aufgetreten; zweitens, die unbekannten Personen treten sowohl im Training als auch in den Tests auf; drittens, die unbekannten Personen treten nur in den Tests auf und werden durch die adversarial Proben ersetzt, die aus den bekannten Proben im Training generiert wurden. Im dritten Fall überschneiden sich die Unbekannten im Training nicht mit den Unbekannten in den Tests. Für die ersten beiden Fälle werden der einfache softmax loss und der entropic open-set loss verwendet, für den zweiten und dritten Fall der objectosphere loss. Wir beweisen, dass diese Modelle eine hohe True Positive Identification Rate liefern, insbesondere wenn die False Positive Identification Rate gering ist. Das Ersetzen der unbekannten Personen durch die adversarial Proben in Fall zwei wie in Fall drei ist ohne Leistungseinbussen erfolgreich. Eine Schwachstelle ist, dass die Eigenschaft der Grössentrennung des entropic open-set loss und des objectosphere loss nicht offensichtlich ist. Bei der Arbeit mit den adversarial Proben ist die Situation noch schlechter.

Contents

Chapter 1

Introduction

Face recognition (FR) has been researched for a few decades. A general face recognition procedure includes face detection and alignment, feature extraction, and score computation (Sáez-Trigueros et al., 2018; Taigman et al., 2014). Over time, the focus of face recognition has shifted from handcrafted features and algorithms to the implementation of Deep Neural Network (DNN). Neural Networks (NN) are end-to-end trainable systems, especially the Convolutional Neural Network (CNN) which is the mainstay in the field of FR and can have human comparable performance on FR tasks (Sáez-Trigueros et al., 2018; O'Toole et al., 2018). CNN is also widely used to construct systems for face and object detection, (Liu et al., 2016; Redmon et al., 2016), and unconstrained age and gender recognition (Levi and Hassner, 2015).

The face recognition tasks started at the restricted closed-set era, that is, every subject appearing in testing (probe set) should have been seen and enrolled into the gallery, and the sample image should be taken within a limited condition. There should not be any surprise in the validation and testing. As the techniques became more advanced, many studies could achieve $> 99\%$ accuracy on the closed-set datasets, the focus turned out to be more unrestricted. Factors like face size, expression, illumination, pose, etc. had more variations. There were also high failure rates in automatic face recognition for several people in one image/video (uncontrolled case) (Beveridge et al., 2013). Since the environment was not experimental anymore and close to real life, the system failed when there were unseen subjects that appeared in the testing. The old models did not learn how to deal with those unknowns and classify them as one of the enrolled subjects. Thus, the research started to work in the open-set era. This system is more close to real-life scenarios since it is unrealistic to get all the people in a general surveillance camera enrolled. There is always someone that we have seen before but are not interested in and someone that appears the first time and we do not care about. The old closed-set FR needs to adapt to the unseen data and be able to reject all uninterested subjects (de O. Cardoso et al., 2017; Günther et al., 2017a,b).

In closed-set identification, softmax solves the separable classification problem; but not enough for discriminative power and generalization (Wen et al., 2016). There are also multiple loss functions designed for the open-set case and we used two of them here. Dhamija et al. (2018) introduced two loss functions, Entropic Open-Set Loss and Objectosphere Loss, and one evaluation metric, Open-Set Classification Rate (OSCR) curve, for open-set object classification tasks to tackle the problem brought by unknown subjects. Math and implementation details are explained later in Chapter 3. The networks are trained with unknown samples. The goal is to magnify the separation of deep feature magnitudes between known and unknown classes. Günther et al. (2020) implemented objectosphere loss in the watchlist problem, which requires the network to detect the faces on the watchlist and ignore the innocents and background. Their work improved the watchlist problem by adding an adapter network trained by the deep features extracted by the VGG2 face recognition network with objectosphere loss. The experiments were established based on a very challenging dataset, UnControlled College Students (UCCS) (Sapkota and Boult, 2013), which usually contains multiple subjects in one image, and images for one subject might be taken in different weather conditions. It has been verified that the objectosphere loss can successfully decrease the deep feature magnitude of innocents and background to 0 and push that of watchlist subjects to the desired value, 5 in their experiment. In the case of the same false alarms per image, adding the shallow network increases the detection and identification rate compared to the original pre-trained model, i.e. performs better. The former is an adapted version of the false alarm rate or false positive identification rate (called in this thesis), and the latter is called true positive identification rate here. Those two quantities are explained in Chapter 3. This implementation has low cost and is easy to generalize to different pre-trained networks and datasets, which leads to the first half of this thesis.

Besides, the adversarial samples could also result in the low performance of the NNs in FR tasks. Adversarial samples refer to inputs with the small but worst perturbations added on (Goodfellow et al., 2015; Rozsa et al., 2016). Those perturbations are subtle to human eyes but cause great damage to the network. So NNs do not assign the generated samples to the original class nor the unknown but give it another label with high confidence. The adversarial samples are usually used to increase the robustness of NNs. In this thesis, we do not use the adversarial images but the generated adversarial deep features to train the models. Further, these adversarial samples do not aim to increase the robustness of the model by classifying them into the known classes but rather work as the unknown samples during the model training. The unknown samples used in the validation and testing are from the subjects that never appear in the training set, that is, no overlapping in the unknown subjects. We want to test the combination of adversarial samples and the introduced loss function in the second half.

In this Master's Thesis, we achieved the following goals: (1) Train a shallow fully connected neural network on the IJB-C dataset (Maze et al., 2018), with the deep feature of galleries as inputs and identity as outputs. (2) Evaluate the Plain Softmax Loss, Entropic Open-Set Loss, and Objectosphere Loss (Dhamija et al., 2018) on the network. (3) Use adversarial samples (Goodfellow et al., 2015; Rozsa et al., 2016) as the unknown samples and evaluate their performance on the Open-Set Face Recognition. (4) Import the trained models into Bob pipeline (Günther et al., 2012) to perform a complete Face Recognition Experiment and evaluate with TPIR (True Positive Identification Rate) vs. FPIR (False Positive Identification Rate) curve, or open-set ROC curve.

In Chapter 2, we briefly review the related work for open-set face recognition. In Chapter 3, we review the mature face recognition experiment tool, Bob, explain in detail the approach to use entropic open-set loss and objectosphere loss in a shallow neural network, and implement the adversarial images as unknown samples on top of that. Also, introduce the methods to apply Bob with the self-trained model. In Chapter 4, we provide an overview of all the experiments and their results. In Chapter 5, we discuss the results and foresee the future work.

Chapter 2

Related Work

2.1 Dataset

The performance of a FR system is highly related to the data. Though constrained FR has high accuracy, as constraints are relaxed and more variations are introduced, accuracy decreases drastically (Beveridge et al., 2013). After solving this problem, unknown subjects are introduced and the accuracy drops again. The datasets summarized below are well-used in the current research.

Labeled Faces in the Wild (LFW) (Huang et al., 2007) dataset contains over 13,000 images and is a popular dataset for experiment benchmark. Each image contains one biggest face with possible variations on views (frontal vs partial-frontal), locations, illuminations, occlusions, and facial expressions. LFW became less challenging as the CNNs involved in the FR. Sapkota and Boult (2013) designed an open-set dataset taken from the surveillance cameras, called UnControlled College Students (UCCS), that currently contains <50,000 images for more than 1,500 subjects (most are known subjects) with variations in pose, illumination, scale, expressions, occlusions, and weathers. Point and Shoot Face Recognition Challenge (PaSC) (Beveridge et al., 2013, 2015) contains still images with different distances to the camera, alternative sensors, frontal vs non-frontal views, varying location, motion blur, and poor focus. It also includes video data and is applied to video person recognition. UMDFaces (Bansal et al., 2017) is a face recognition dataset with 367,888 annotated faces of 8,277 subjects from the unconstrained videos. It also contains pose variations. Ms-Celeb-1M (Guo et al., 2016) is a dataset for large-scale face recognition, 100k celebrities with 10 million images. These samples are grabbed automatically from the internet. The benchmark (CNN model) performance for celebrity FR task has close to human behavior. VGGFace2 (Cao et al., 2018) dataset contains 9,131 subjects and each subject has more than 300 samples on average. As the other large-scale FR dataset, though it contains variations in pose, age, illumination, ethnicity, and profession, training with VGGFace2 improves the performance on age and pose-related tasks.

Yi et al. (2014) built a large-scale dataset called CASIA-WebFace, which includes around 10,000 subjects with 500,000 samples taken in the wild. Although this dataset is collected from the Internet, it is not overlapped with LFW. This is only for training CNNs purpose. Unconstrained face recognition dataset IARPA Janus Benchmark A (IJB-A) (Klare et al., 2015) has a mix of images and videos from 500 subjects with full pose and geographic variations. All faces have hand-labeled bounding box information but only a few of them include locations of eyes. They are not filtered by a commodity face detector. More importantly, IJB-A contains protocols for open-set face identification and verification. CNN has been proved to perform better than traditional methods on the IJB-A dataset (Chen et al., 2016). Bilinear CNN also has good performance on the IJB-A (Lin et al., 2015; Chowdhury et al., 2016). CNN-based triplet probabilistic embedding shows the robustness with IJB-A dataset (Sankaranarayanan et al., 2016). IARPA Janus Benchmark B (IJB-

B) (Whitelam et al., 2017) dataset is a superset of IJB-A. It contains all variations mentioned in IJB-A but more uniform geographic distribution subjects (1,845) and samples (21,798 still images and 55,026 frames from 7,011 videos). Its test protocols are appropriate for open-set face identifications in environments like an access point and surveillance video. This thesis is built on the dataset IARPA Janus Benchmark C (IJB-C) (Maze et al., 2018), which is a superset of IJB-A and IJB-B datasets. The details are explained in section 4.1. MegaFace (Kemelmacher-Shlizerman et al., 2016) dataset includes 690k subjects and 1M samples with increasing numbers of "distractors" in the gallery. It is a benchmark of a million faces, which is closer to the real situation. They discover that the experiments on a large-scale dataset exhibit the discrepancy on algorithms easily. IJB's and MegaFace are designed for evaluation and building the benchmark for CNN models.

Besides, WIDER FACE (Yang et al., 2016) is a face detection dataset and 10 times larger than existing face detection datasets. It creates a training environment that is close to the real-world situation with annotation provided, variations in scale, extreme pose, and occlusions. Most face detection models have a low performance with this dataset because of the above-mentioned variations.

2.2 Models and Algorithms

We specifically focus on face recognition with deep features. First, an end-to-end network is trained on large datasets and the output layer uses the softmax activation. Combing with the softmax outputs, we can use the cross-entropy loss function to calculate the loss for multiple classes. Second, the last layer of the network is removed. Third, we pass the images from the previously unseen people into the network and the updated output layer provides their deep features. Then, those deep features are compared by some distance functions and we assign them a label according to the scores. Thus, the dataset used for training the network has no overlap with the one for the evaluation in step three. The evaluation datasets are split into two parts, the deep features of the faces in the first part are enrolled into a gallery, and the images in the second part work as the probe. Comparison happens between the gallery and probe. The probe set without unseen subjects composes the close-set experiment, and the one with some unknown people gives the open-set experiment.

2.2.1 Closed-Set Era

Since in the unconstrained cases, the intra-class variations of deep features increase and challenge the models that are trained with limited variations, the goal is to minimize the intra-class variations as well as maximize the inter-class variations in the deep feature space. Deep IDentification-verification features (DeepID2) is a CNN model designed to extract the deep features that strive to achieve this goal (Sun et al., 2014). DeepFace (Taigman et al., 2014) uses explicit 3D face modeling to align face and extract features and trains on a large dataset with enough samples per subject. It achieves a human-level face verification performance on the LFW dataset. A single 11-layer CNN built by Yi et al. (2014) and trained with CASIA-WebFace outperforms DeepFace and DeepID2. Early supervision is implemented in DeepID2+ (Sun et al., 2015), which also increases the dimension of features to get a good performance on LFW and Youtube Face (YTF) Dataset (Wolf et al., 2011). The hidden neurons of DeepID2+ are highly but selectively active for different identities, which is similar to our goal for the deep feature responses on known and unknown identities. DeepID2+ also reported a relatively good performance on the open-set face identification task, but this net is not specifically designed to resolve the open-set problem.

PCANet (Chan et al., 2015) is a simple structure composed of cascaded principal component analysis (PCA), binary hashing, and block-wise histograms. It builds a comparable result on FR tasks with different datasets, but PCANet is not able to deal with the difficult variations in Pascal (Everingham et al., 2010) and ImageNet (Deng et al., 2009) due to its simplicity. FaceNet (Schroff et al., 2015) does not use the representation created by the intermediate bottleneck layer (so-called the deep features here) and rely on those representations to generalize the model. Instead, it directly maps face embeddings for each image into Euclidean space for face recognition purposes with the corresponding squared distance as the similarity score, a smaller score is preferable. This embedding is applicable for large-scale datasets efficiently. The system uses triplet loss.

Residual Network (He et al., 2016) improves the performance of image recognition since it is deeper but less complex and easy to optimize. It can also be adapted to the face image descriptor for low-quality surveillance camera samples (Herrmann et al., 2016a). Combining CNN with triplet probabilistic embedding (Sankaranarayanan et al., 2016) is robust for the extreme pose variation and saves the training time. A CNN with a manifold-based track comparison strategy is applied to the low-resolution problem from the surveillance camera (Herrmann et al., 2016b). This approach makes the model to be noise-resistant and outperforms VGG-Face (Parkhi et al., 2015). In addition to the network, a loss function called center loss (Wen et al., 2016) is introduced to enhance the discriminative power of the model by updating the class centers and penalizing the distance between deep features and the corresponding class centers.

2.2.2 Open-Set Era

Open-set recognition is more close to the real scenarios, where the system should be able to identify the subjects that we are interested in, as well as reject the uninteresting ones (de O. Cardoso et al., 2017). Experiment shows that good algorithms perform poorly in open-set databases. Bendale and Boult (2016) introduced a new layer called OpenMax to resolve the open-set problem in the view of the model framework. It is an alternative to the softmax function, the last layer of the network, and uses the values from the activation function of the penultimate layer and calculates its probability to be an unknown.

It has been proved that thresholding similarity scores cannot reflect the performance of open-set FR models, but the Extreme Value Machine (EVM) method, the one that is derived from statistical Extreme Value Theory and flexibly adapts to the feature space of unseen subjects (Rudd et al., 2018), performs well in evaluations of both close- and open-set cases (Günther et al., 2017a; Dhamija et al., 2018). Günther et al. (2017b) evaluated the performance of different face detectors and recognition networks on the UCCS dataset and further proved that they have a good performance on face verification or closed-set face identification, but not on the open-set face identification. Entropic Open-set Loss and Objectosphere Loss are introduced to tackle the effects from the unknown samples. Both loss functions can separate the deep feature magnitudes for knowns and unknowns (Dhamija et al., 2018; Günther et al., 2020).

Angular softmax (A-Softmax) loss, introduced by Liu et al. (2017), is specifically designed for open-set FR tasks. It learns angularly discriminative features and penalizes the angles so that the intra-class distance is more compact than the inter-class distance. As an extension of increasing discriminative power and using angular margin, large margin cosine loss (LMCL) (Wang et al., 2018) is proposed. It utilizes the L_2 normalization and cosine margin to remove radial variations and maximize margin in the angular space. CosFace is the typical model trained with LMCL. The research on loss function focuses on using margins into the general loss functions to obtain a better performance on face recognition tasks. In this thesis, we utilize the model ArcFace (Deng et al., 2019), a deep face recognition model with a novel Additive Angular Margin Loss. It is a penalty term for the angle between deep features and target, where arc-cosine is used to calculate the angle and an additive angular margin is added to the target angle before forwarding to the cosine distance. This calculation only has a subtle extra cost but increases the discriminative power efficiently.

2.3 Adversarial Attacks

In this thesis, the generated adversarial samples involve in the model training as the unknown samples. We usually make small modifications, or noises, to the original images to generate the adversarial samples. Those differences are tiny but have a great impact on the classification results for the neural networks. We use two fast methods, Fast Gradient Sign (FGS) (Goodfellow et al., 2015) and Fast Gradient Value (FGV) (Rozsa et al., 2016). The former is introduced by Goodfellow et al., who think that the neural networks are not robust to the adversarial samples because of their linearity and emphasize the importance of perturbation direction. The latter is created by Rozsa et al. and has similar mechanics to FGS but with an improvement in the quality of the samples. The detailed math and application are explained in Chapter 3. Psychometric Perceptual Adversarial Similarity Score (PASS) measure is introduced to quantify the imperceptible perturbations and generating hard positives gives a new direction for adversarial images (Rozsa et al., 2016).

Chapter 3

Approach

The general face recognition experiment refers to a complete process: a dataset with a purpose-specific protocol is sent into face detection, alignment, and feature extraction. The extracted features of the gallery G are enrolled and the probe P is compared to the former and computes the similarity score to decide the subject identity. This process is an extended version with the evaluation procedure for the one in section 2.2 since it includes the details for preprocessing, skips the model training, and uses the pre-trained model to extract features directly. Similarly, open-set FR in this process refers to the unseen subjects in the probe. The enrolled gallery subjects are represented by "knowns" and we use K for the known class set, and unseen subjects are "unknowns" and we use U to represent the unknown class set.

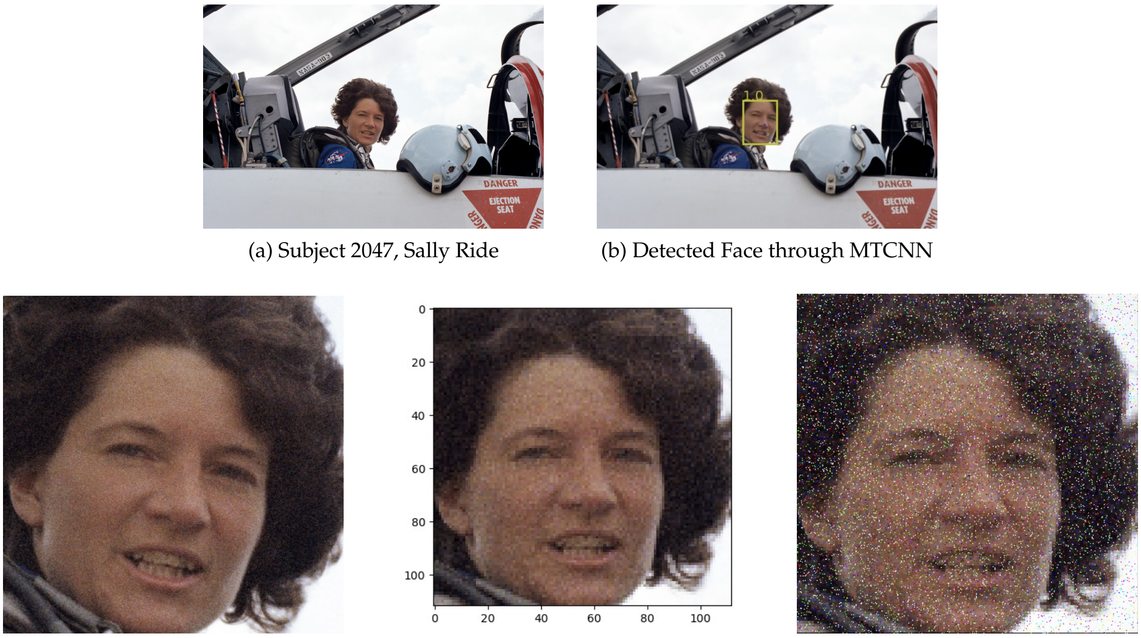
The performance of open-set FR highly depends on the face detection and feature extraction steps. We focus on the latter and try to train a shallow fully connected neural network to enlarge the difference between the knowns and unknowns, i.e. we use the deep features extracted by a deep neural network as the input to train a shallow network that could split K and U better. Two sources of known unknowns are applied in the model training, the subjects that never appear in the knowns, and the generated adversarial samples for the knowns. The first source of known unknowns is used as unknowns in the validation and testing. Thus, if training with the first source, there is no unseen subject in the validation, and if the second source involves, then the unknown subjects in the validation have no overlap with that in the training. This network is added back as an extra feature extractor to the FR experiment for evaluation.

3.1 Face Recognition Pipeline

The face recognition experiments are built based on the framework provided by the open-source toolbox Bob. As it is hard to reproduce the face recognition research with a business purpose, their outcomes do not contribute to the improvement of the open-source scientific research (Günther et al., 2012). The Biometric Security & Privacy Group at Idiap Research Institute built Bob for the signal processing and machine learning researches (Anjos et al., 2012).[1] `bob.bio.base` and `bob.bio.face` are two packages contained in Bob and are mainly used here for the construction of face recognition experiments. `bob.bio.base` is the base package that defines the structure of the biometric recognition experiment and those structures are specified and adapted to the specific purpose of use.[2]

[1] https://www.idiap.ch/software/bob/

[2] https://www.idiap.ch/software/bob/docs/bob/bob.bio.base/stable/index.html



(a) Subject 2047, Sally Ride

(b) Detected Face through MTCNN

(c) Magnified original face

(d) Face Crop and Align the eye location

(e) Adding Noise

Figure 3.1: Example Face Detection, FaceCrop, Alignment, and Adding Noise. The image is from IJB-C (Maze et al., 2018) dataset and the female in (a) is Sally Ride, subject 2047. (b) to (e) are only for illustration purposes and pass into the preprocessing steps separately.

Dataset Protocols & Annotations

`bob.bio.face` is designed for the face recognition experiments and it contains the multiple face recognition tools, traditional or deep learning, for database, preprocessor (face detection and alignment), feature extractor, and algorithm (score calculation).[3] Each dataset has at least one protocol defined in the `bob.bio.face.database`. According to their sources, each dataset has different annotations, such as `"eye-centers"` or `None`. Face detection, or an annotator defined in package `bob.ip.facedetect`,[4] is applied when the default annotation is `None` or does not satisfy our requirement for alignment. Our dataset is IJB-C (Maze et al., 2018), which is explained in detail in Section 4.1, has default annotator `"bounding-box"`, means it only provides the top-left and bottom-right coordinates of the face.

Preprocessor

The protocols provide a bounding box for the faces we care about in each image. Assume there is only one bounding box in each image, it is necessary to remove the noises caused by non-main people and background. So the face is first cropped according to the bounding box. Then, we choose to use Multi-task Cascaded Convolutional Networks (MTCNN) (Zhang et al., 2016)[5] and its implementation on Bob for face detection and each image, which returns the following coor-

[3] https://www.idiap.ch/software/bob/docs/bob/bob.bio.face/stable/index.html

[4] https://www.idiap.ch/software/bob/docs/bob/docs/stable/bob/bob.ip.facedetect/doc/index.html

[5] https://kpzhang93.github.io/MTCNN_face_detection_alignment/index.html

dinates for the face contained in the bounding box: `topleft, bottomright, reye, leye, nose, mouthright, mouthleft, quality`.[6] Figure 3.1(a) is an image from IJB-C and Figure 3.1(b) is the detected face with landmarks. Notice that this example is for illustration purposes only and not involves in the normal workflow, i.e. the face is not cropped before the face detection.

The cropping is incapable to remove the noises caused by non-frontal pose, occlusion, and facial expression, as shown in Figure 3.1(c). The detected landmarks from above are used to align the face to the desired position, for instance, the eyes should be symmetric, before passing into the models for extracting the features of each face. `bob.bio.face.preprocessor.FaceCrop` is designed for this purpose.[7]

As shown in 3.1(c), if we crop the original image according to the bounding box, only the lady's face is left. If we define a standard face to be completely frontal and has symmetric eyes in the same horizon, then Sally Ride's face is not in a standard position, because her nose is not perpendicular to the horizontal line, and if we use a line to connect her eyes, this line is not parallel to the horizontal line. Our goal is to align her face so that the landmarks for eyes are in a standard format and can be passed into the feature extraction step. Figure 3.1(d) is a face crop example according to the detected landmarks shown in (b). We mainly rely on eye positions. Given the desired positions (usually in the upper part) of eyes, the system aligns the detected eyes into those positions which affect the other landmarks on her face. Now, her eyes are horizontally symmetrical, same for her mouth, and her nose is perpendicular to the horizontal line. Face cropping and alignment ensure that no image background is passed into the next step; no unnecessary information is recognized as the feature of the face; and in all images, eyes are in the same position.

Extractor

Bob imports many pre-trained face recognition neural networks in this step for the feature extraction. The preprocessed images are forwarded into the neural network and the outputs are the deep features in tensor format. The framework is flexible so that we can use our pre-trained neural network here or apply multiple networks at the same time. We choose to use the MxNet framework with the default ArcFace Resnet100 backbone model as the baseline model (Deng et al., 2019).[8] Further, we train a shallow fully connected neural network as mentioned above and add it after implementing the ArcFace model. The outputs with dimension 512 from the ArcFace model are the inputs for the shallow network, and the outputs of this network are the identities. The penultimate layer of the network is the deep features that are passed into the score calculation and evaluation. The details of this network are explained in Section 3.2.

Algorithm & Evaluation

All of the above steps are applied on all the data, i.e. for both gallery set and probe set. The separation of sets is only worth discussing in score calculation and result evaluation. Features of samples from the gallery are enrolled with their corresponding subject ID, and then samples from the probe set will be compared with the former by calculating their similarity scores. The scoring function can be called in package `bob.bio.base.pipelines.vanilla_biometrics.Distance`.[9] The default is to calculate cosine distance by `scipy.spatial.distance.cosine`. Those scores are used to plot an evaluation curve.

[6] https://www.idiap.ch/software/bob/docs/bob/bob.ip.facedetect/stable/mtcnn.html

[7] https://www.idiap.ch/software/bob/docs/bob/bob.bio.face/stable/implemented.html#bob.bio.face.preprocessor.FaceCrop

[8] https://www.idiap.ch/software/bob/docs/bob/docs/stable/bob/bob.bio.face/doc/implemented.html

[9] https://www.idiap.ch/software/bob/docs/bob/bob.bio.base/stable/py_api.html#bob.bio.base.algorithm.Distance

We use the baseline configuration provided by Bob to run the FR experiment and modify it to apply the new feature extractor in section 3.2 and similarity score function in section 3.3.3. The baseline `arcface-insightface` uses `MTCNN` for face detection, `FaceCrop` for preprocessing, `MxNet` framework with ArcFace Resnet100 backbone model for feature extraction, and cosine distance for scoring.[10]

3.2 Network Training

3.2.1 Fully Connected Neural Network

We construct a shallow fully connected neural network. There are three fully connected layers, with the first one followed by the activation function. As shown in Figure 3.2, the inputs (red) linearly forward to the first hidden layer (blue), followed by an activation function (brown), then linearly forward to the second hidden neurons (green), and then the outputs (pink). The inputs are the deep feature of the face in each sample image extracted from the ArcFace Resnet100 (Deng et al., 2019), and they have dimension 512. The outputs are the number of known subjects in K. The second hidden layer is also the deep features extracted by this network.

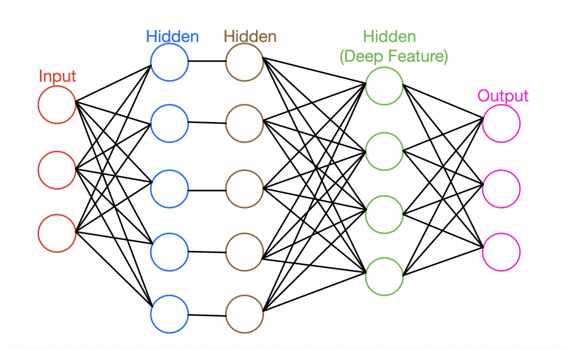


Figure 3.2: Sketch of Shallow Neural Network.

3.2.2 Loss Function

We implement three loss functions in the shallow neural network training.

Plain Softmax Loss

Let $k \in K$ represent the known class, $ku \in U$ represents the known unknowns, $uu \in U$ represents the unknown unknowns. The output values from the shallow neural network are the predictions by matrix multiplication of weights and the last hidden layer (green), i.e. logits. The logits, represent by z_k for k^{th} class, are used for calculating the Standard Softmax Score and the corresponding Cross-Entropy Loss (Plain Softmax Loss)

$$S_k(x) = \frac{e^{z_k}}{\sum_{k' \in K} e^{z_{k'}}}, \tag{3.1}$$

[10] https://www.idiap.ch/software/bob/docs/bob/bob.bio.face/stable/baselines.html#deep-learning-baselines

$$J_{CE}(x) = -\sum_{k \in K} \mathbb{1}(x = k) \log S_k(x), \tag{3.2}$$

where x is a sample. This is implemented in the PyTorch and can be called easily.

Entropic Open-Set Loss

The entropic open-set loss function J_E is a derivation of plain softmax and aims to classify unknowns U from knowns K by maximizing the entropy of the unknown samples (Dhamija et al., 2018). If the sample is in class $k \in K$, then J_E will keep the plain softmax results; otherwise, J_E tries to equalize the logit value for samples in $u \in U$ because unknowns should not have a preference to any known subject.

$$J_E(x) = \begin{cases} -\log S_k(x) & \text{if } x \in k \\ -\frac{1}{|K|} \sum_{k \in K} \log S_k(x) & \text{if } x \in u \end{cases} \tag{3.3}$$

Objectosphere Loss

The objectosphere loss function J_O is an improvement on the entropic open-set loss function to magnify the feature magnitudes separation between known and unknown classes (Dhamija et al., 2018). Both logit values and features are involved in the calculation, where the latter is used to constrain the deep feature magnitudes. If the sample is from known class $k \in K$, then J_O tries to push the magnitude of the sample's feature to be at least ξ; otherwise, J_O penalizes the feature magnitude and pushes it to be zero. Thus, for $x \in k$, we have lower entropy and larger features around ξ, vice versa. In equation (3.4), α value represents the power of magnitude constrain to J_E.

$$J_O(x) = J_E + \alpha \begin{cases} \max(\xi - \|S_k(x)\|, 0)^2 & \text{if } x \in k \\ (\|S_k(x)\|)^2 & \text{if } x \in u \end{cases} \tag{3.4}$$

The implementation of entropic open-set loss and objectosphere loss refers to the GitHub page of Vision And Security Technology (VAST) Lab.[11]

3.2.3 Model Training

The models are trained with different training sets but the same validation and testing sets. All experiments view subjects in gallery $G1$ from the IJB-C dataset (Maze et al., 2018) as the knowns $k \in K$. In the case that using images from gallery $G2$ as unknown samples in the training, we call them known unknowns $ku \in U$. When using adversarial samples as unknown samples in the training but gallery $G2$ in the validation and testing, we call adversarial samples known unknowns and subjects in gallery $G2$ as unknown unknowns $uu \in U$. The testing set contains only the probe samples provided by the dataset IJB-C. Subjects from both $G1$ and $G2$ are included in the probe set. The validation set is composed of one sample per subject from $G1$ and $G2$ if there is more than one sample per subject. The validation and testing sets are fixed and shuffled before implementation. The base model is trained only by the known samples $x \in k$ with plain softmax loss and tested by the probe set.

[11] https://github.com/Vastlab/vast/blob/main/vast/losses/losses.py

Known Unknowns

We call subjects in $G2$ as known unknowns ku because these subjects are involved in model training, validation, and testing but with label -1 instead. They are randomly shuffled together with the knowns k before being forwarded into the neural network. Thus, there are no unseen subjects in validation and testing sets.

Adversarial Images

The second part of this thesis is to test the performance of different loss functions when there are unknown unknowns. In this case, we use knowns k and the generated adversarial samples in training instead. The adversarial samples do not appear in either validation or testing. Figure 3.1(e) is an example of generating the adversarial image. The following two approaches are applied to generate the adversarial samples:

(a) Fast Gradient Sign (FGS)

$$\hat{X}_{FGS} = X + \epsilon \text{sign}(\nabla X), \tag{3.5}$$

where ∇X is the gradient of input X with respect to the loss J, the plain softmax loss, and ϵ is the attack step size, the smaller ϵ means making fewer perturbations on the original X. FGS takes the ϵ value as all perturbations, and the direction depends on the sign of ∇X, while the direction of perturbation is the most important factor in generating adversarial samples. The perturbation is a dense random noise but evenly spread among the entire image (Goodfellow et al., 2015).

(b) Fast Gradient Value (FGV)

$$\hat{X}_{FGV} = X + \epsilon \frac{\nabla X}{\max(\|\nabla X\|)} \tag{3.6}$$

Instead of only relying on ϵ, FGV scales the gradient value by dividing by the maximum among all of them. Thus, the magnitude of the gradient also affects the perturbations on the original X. Compared with FGS, FGV creates more local perturbations but also efficiently affects the classification performance (Rozsa et al., 2016).

The implementation of FGS and FGV refers to advertorch, which is a Python toolbox for adversarial robustness research (Ding et al., 2019).[12] In each training epoch, the model is first trained on the known samples from k, and only when the softmax value of a sample is greater than a threshold, the adversarial attacks are applied on that sample, which ensures that the model has a great probability to identify that sample correctly before generating adversarial samples.

Since the input X in our experiments is not an image but its deep feature, it is necessary to make some adaptations to the given implementation. Each batch of $x \in k$ for training purposes is forwarded into the network as usual. When using FGS, after the backpropagation, the sign of the gradient is retained and multiplied by the step size ϵ, which is composed by the product of two values, finally adds back to x to create the adversarial samples. For the FGV case, the entire gradient is retained and divided by its maximum before multiplying ϵ. Compared with the setup in the GitHub,[12] we remove the re-computation of the gradient to not reset the gradient back to the start point, and not clamp adversarial samples to the range $[0, 1]$, since it is not appropriate for the face features.

Fine-tuning ϵ is necessary to find a good separation and/or open-set ROC curve. ϵ is defined to be a product of a value between 0 and 1 and the absolute maximum among each input, thus a different ϵ is calculated for each sample. The absolute maximum among all inputs could be a very large value and only appears a few times in the entire dataset. For a random input, if the

[12] https://github.com/BorealisAI/advertorch/blob/master/advertorch/attacks/one_step_gradient.py

absolute maximum of the dataset is much larger than that of this input, then it is incapable to grab the specialties for this sample. The product with the larger one brings too much change to the original input, which deviates from our goal to make the model be able to identify the subtle changes. Similarly, the first value in the product also determines the potential differences between original and generated samples. A constant value or a decaying value as in Equation (3.7) can be applied. A constant value like 0.9 is too large so that the product ϵ is large and results in an adversarial sample that is too different from its original value. This is meaningless because the network still cannot identify the small changes. Conversely, 0.001 is too small so it takes infinitely many epochs to train the network. Decaying with epoch implies that the ϵ is close to the lower bound 0.01 as the network keeps training, and thus the adversarial samples are closer to the true input and harder to identify.

$$\epsilon = \max((0.95)^{epoch}, 0.01) * |\max(x)| \tag{3.7}$$

3.3 Evaluation Metrics

3.3.1 Feature Magnitude Visualization

Training the shallow neural network with entropic open-set loss and objectosphere loss is aimed to separate the feature magnitudes of knowns and unknowns. So we use the density plot of the feature magnitudes to evaluate the separation performance of models, as shown in Figure 4.3. We expect that when using the original features, knowns and unknowns have very similar distributions and large areas of overlap. The entropic open-set loss is able to shift the distribution of unknowns to the left, that is, reducing the magnitude of unknowns and the overlapping area with the knowns. Objectosphere loss strives to intensify this left shift and pushes the magnitude of knowns to the size ξ that we want (Dhamija et al., 2018; Günther et al., 2020).

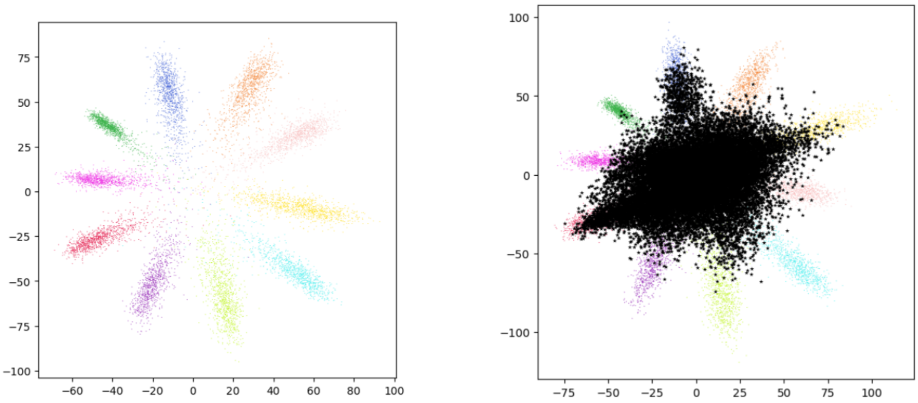
3.3.2 Confidence Measurement & Area Under the Curve (AUC)

The class number defined in the neural network output is $|K|$, and we do not include a class for the unknowns. So accuracy measure does not perform well in this case because not all labels are told in advance (Dhamija et al., 2018). Thus, the confidence measure is applied. Confidence is the standard softmax value for the desired class if the sample is from known class $k \in K$; otherwise,

$$confidence = 1 - \hat{S_k} + \frac{1}{|K|} \tag{3.8}$$

, where $S_k(x) = \frac{e^{z_k}}{\sum_{k' \in K} e^{z_{k'}}}$ is the softmax value for class k, and $\hat{S_k}$ stands for the maximum softmax value among all classes and $|K|$ stands for the number of known classes. Here, since the maximum of $\hat{S_k}$ should be $\frac{1}{|K|}$, the last component is added to offset its effect with the fact that the maximum of confidence is 1.

When the adversarial samples are used to train the network, the situation is different. We keep using the confidence measure in the training evaluation but it is not appropriate for the validation. Schnyder (2021) investigated the feature space of MNIST handwritten digits database (LeCun, 1998). Samples in the MNIST work as the known samples and samples of handwritten letters from EMNIST (Cohen et al., 2017) as the unknowns. By setting the dimension of the deep features, they can be plotted in a two-dimensional space, as shown in the illustration plot Figure 3.3. In this plot, each class of MNIST digits has a different color and looks like a petal of a flower. But the deep features of unknown EMNIST letters are in black and overlap with the known petals



(a) MNIST only

(b) MNIST and EMNIST letters

Figure 3.3: Feature space of MNIST and EMNIST databases from Schnyder (2021). MNIST samples work as knowns and each class is colored differently. EMNIST samples are the unknowns and depicted in black.

to a great extent. The deep features for the faces have higher dimensions but the overlapping of unknowns and knowns are similar. When training with adversarial samples, the feature space for unknowns is more spread than the known unknown case, and the feature space for knowns is correspondingly shrunk. As a result, the confidence for an unknown sample is higher and causes the unreliability of confidence measurement.

Using the Receiver Operating Characteristic (ROC) Curve and then computing the Area Under the Curve (AUC) becomes the evaluation metric for the shallow neural network. ROC is used when we want to evaluate the performance of binary classification. ROC is a true positive rate (TPR) against false positive rate (FPR) plot. By comparing the softmax values for knowns and unknowns against the real class labels, TPR is the rate that a known sample is classified correctly and FPR is the rate that an unknown sample is incorrectly classified as the knowns. AUC score is the area under the ROC curve. It expects to return a value between 0 and 1, the higher the value, the better the prediction performance. We skip the plotting of the ROC curve but only use the AUC to evaluate the network. We also attempt the ROC by comparing the deep feature magnitudes for knowns and unknowns against the real class labels.

3.3.3 Similarity Score

The pre-trained shallow neural network is imported into the Bob pipeline, right after the ArcFace model in the feature extraction step. Then for all extracted features, we calculate the similarity score between features from the gallery and the probe. When there is no obvious magnitude separation for knowns and unknowns, general cosine similarity (3.10) is applied; otherwise, weighted cosine similarity (3.11) could significantly improve the evaluation results because the feature magnitudes are modified through model training (Günther et al., 2020).

$$cdist = 1 - \frac{a^T b}{\|a\|_2 \|b\|_2}, \tag{3.9}$$

$$Cos(Scores) = 2 - cdist, \tag{3.10}$$

$$WCos(Scores) = (2 - cdist) * \|F\|_2, \tag{3.11}$$

where a and b are the samples from the gallery and from the probe, respectively, and F is the sample from the probe set (F is b here). Since `cdist` changes the original cosine similarity domain $[-1, 1]$, 1 for best, to $[0, 2]$, 0 for best, $2 - cdist$ converts the domain to $[0, 2]$ but 2 for best, which is consistent to our commonsense. Furthermore, when there are multiple samples for the same subject involved in the calculation, for instance, the IJB-C dataset makes the multi-image template (Maze et al., 2018), we choose the default method to compute the weighted average of all samples and forward it to the score calculation. No matter how many samples for each subject the template provides, only the weighted average of them is used. This is also the default approach defined in the IJB-C dataset and we stick to it. It might be different to not use the average over all samples but calculate the scores for each sample then do some different averaging methods. This is just a random guess and we will not further discuss it here.

3.3.4 Open-set Receiver Operating Characteristic (ROC) Curve

Given the cosine similarity scores, an open-set ROC curve (OSCR curve is an adaptation but quite similar) is made to visualize the FR results, as shown in Figure 4.4. It is a True Positive Identification Rate (TPIR) against False Positive Identification Rate (FPIR) plot. TPIR, equation (3.13), is the number of samples that are correctly classified and their similarity score to that class is equal or above a threshold θ over the total number of known samples. FPIR, equation (3.12), is the number of unknown samples that are classified as one of the known class k over the total number of unknown samples (Phillips et al., 2011). The implementation of open-set ROC plot is given in package `bob.bio.base`.[13]

$$FPIR(\theta) = \frac{\{x | x \in u \wedge max_k P(k|x) \geq \theta\}}{|U|} \tag{3.12}$$

$$TPIR(\theta) = \frac{\{x | x \in k \wedge \operatorname{argmax}_k P(k|x) = \hat{k} \wedge P(\hat{k}|x) > \theta\}}{|K|} \tag{3.13}$$

[13] https://www.idiap.ch/software/bob/docs/bob/docs/master/bob/bob.bio.base/doc/biometrics_intro.html#evaluation

Chapter 4

Experiment

4.1 Datasets

The entire experiments and evaluations are based on the dataset IJB-C (Maze et al., 2018), which is a superset of above mentioned IJB-A and IJB-B datasets. IJB-A, B, and C are designed for unconstrained face recognition research. They include subjects that are more general and less occupationally and geographically specific than the other datasets with full variation in pose and occlusions. Faces are manually labeled. Although there might be multiple faces in one sample image or frame, only the labeled face, given by the protocol, is viewed as the subject and used for face detection alignment (Klare et al., 2015; Whitelam et al., 2017; Maze et al., 2018). IJB-C extends IJB-B to 3,531 subjects, which split into two disjoint galleries $G1$ (1,772 subjects with 5,588 samples) and $G2$ (1,759 subjects with 6,011 samples). For enrollment purposes, the samples in galleries are still images that have better resolution than frames. We utilize the 1:N end-to-end mixed protocol with 31,415 probe samples, which are a mix of still images and frames. Given different protocols, the IJB-C dataset generates different templates. The 1:N mixed recognition protocol generates multi-image templates, that is, there are multiple subjects and each subject has multiple samples contained in the template. These samples could either be still images or frames. When we want to do the evaluation with the multi-image template, the deep features of each subject are defined as the weighted average over all samples. Since the quality of frames is lower than that of the still images, less weight is distributed to the frames and the combined weight of all the frames is equal to the weight for one still image.

4.2 Basic Setup

The experiments aim to evaluate the performance of entropic open-set loss and objectosphere loss (Dhamija et al., 2018) on the IJB-C dataset, and whether generating adversarial samples as known unknowns (Schnyder, 2021) could be applied with those loss functions as well. We follow the default open-set protocols for the IJB-C dataset. All the samples in this protocol, in gallery $G1$ and $G2$ and probe P, are put into the bob pipeline for face detection by MTCNN, alignment by face crop, and feature extraction by ArcFace InsightFace R100 (Deng et al., 2019).[1] Usually, the next step is to separate the features and calculate the score.

We add one more step in between the feature extraction and the score calculation. A shallow fully connected neural network, with the deep features as the inputs and their corresponding identities as the outputs, is trained, first by using samples from $G2$ as known unknowns and

[1] https://github.com/deepinsight/insightface/tree/master/model_zoo

employing entropic open-set loss and objectosphere loss and then using adversarial samples as known unknowns with objectosphere loss. Besides, the adversarial case is also trained with entropic open-set loss but eventually discarded. Because objectosphere loss is basically the extension of open-set loss and expects to have better performance, we choose to jump to the objectosphere loss directly. Finally, as shown in Figure 4.1, we run the complete FR procedure with the ArcFace extractor followed by a trained model, where the last layer of the trained network has been removed so that it returns the deep features in the penultimate layer, as an extra extractor to evaluate its performance and compare the results with the pure ArcFace results.

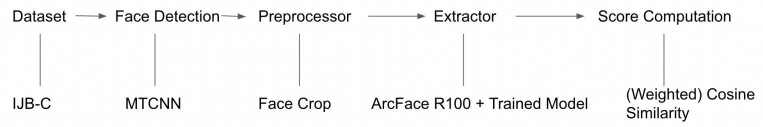


Figure 4.1: Illustration for a complete FR experiment.

4.3 Shallow Model

Only gallery $G1$ is labeled as the known sample set, $G1 = K$, and we define subject k in $G1$ as the known subject. The known unknowns differ for each experiment. The model outputs are chosen to be the number of logits that equals the number of known subjects, in this case, 1,772. The training set, validation set, and testing set are set up as defined above. Early stopping is applied to train most of the networks, we point out cases where it is not used. The training will stop if no improvement has been made for more than 1,000 epochs. Only the samples used in the base model training are balanced and for the rest, we do not attribute any weight to the samples, so both known samples and unknown samples have the same weight. We have a summary table, Table 4.2, for the best version of all four models with the corresponding parameters and evaluation measurement listed.

Base Model

We define the base model by not using known unknowns in the model training and validation. Thus, $G1$ is split into training and validation. As described above, the network has three fully connected layers with one activation $tanh$ and is trained by softmax entropy loss. In Table 4.1, our possible model parameters are listed. The first two columns are one-to-one corresponded since we separate each model by the known unknowns and its loss function. Columns 3 to 6 are the choices for each parameter. We cross multiply columns 3 to 6 and select a subset of all combinations. Some parameters like network layer size 64by32, are proved to be useless regardless of other parameters. This is discussed in Chapter 5.

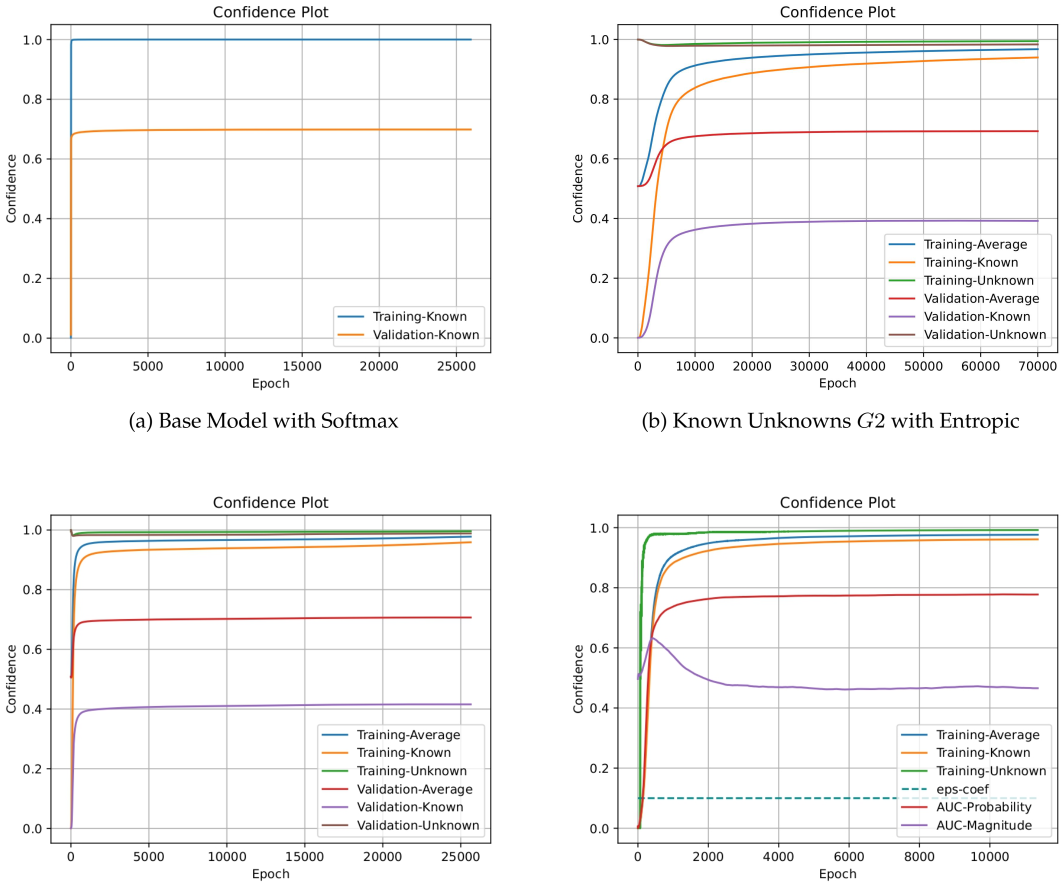
The confidence for recognizing the identities is shown in Figure 4.2(a). Classes are balanced before training. When choosing SGD as an optimizer with a learning rate of 0.02, the confidence approaches the plateau fast, while we only get about 0.6987 in the validation. As a comparison, another splitting method is applied. For each known class, if it contains more than five samples, then 20% of them are moved to the validation set and the rest stay in the training set; if the number of samples is greater than one, then one of them belongs to the validation; otherwise, this subject is not in the validation set. This splitting reaches a 0.7983 confidence. This is expected since the model is evaluated more often on a subject that has more samples in the validation set, and thus results in higher confidence. Figure 4.3(a)-4.3(c) exhibit the feature magnitude for the original inputs and 4.3(d)-4.3(f) for the base model case with the best combination of parameters listed in Table 4.2 column 2. They are all normalized to have a standard height. Our goal is to compare the magnitude for knowns and unknowns when no or different unknowns join the model training. Complete overlapping happens if we do nothing on the original features, and softmax entropy loss only contributes to the separation slightly. But the base model also pushes the dominant proportion of magnitude to the 50-60 range, instead of falling in the 20-25 range as for the original features.

Model	Loss Function	Network Layer Size fc1&fc2	Optimizer w/ lr	Normalization	Dropout
Base Model Known Unknowns $G2$ Case1 Known Unknowns $G2$ Case2 Adversarial Samples	Softmax Cross Entropy Entropic Open-set Loss Objectosphere Loss Objectosphere Loss	2048by1024, 1024by512, 128by64, 64by32	SGD(0.02), Adam(0.0001)	YES, NO	YES, NO

Table 4.1: Common parameters to choose for all the models. Model and Loss Function is 1-1 corresponded, columns 3 to 6 are the options for each parameter that we use to train the model. *Network Layer Size fc1&fc2* stands for the number of neurons in the first and second hidden layers. *Normalization* is **YES** when the inputs are normalized before forwarding into fc1. *Dropout* is **YES** when dropout is applied in between fc1 and fc2.

	Base Model	Known Unknowns $G2$ Case1	Known Unknowns $G2$ Case2	Adversarial
Training Unknown Source	No Unknown	Gallery G2	Gallery G2	Adversarial Samples
Loss	Plain Softmax Loss	Entropic Open-set Loss	Objectosphere Loss	Objectosphere Loss
Network Layer Size	2048by1024	1024by512	2048by1024	1024by512
Weight	YES	NO	NO	NO
Normalization	NO	NO	YES	YES
Dropout (After fc1 & activation)	NO	NO	NO	NO
Optimizer	SGD	SGD	Adam	Adam
Learning Rate	0.02	0.02	0.00001	0.00001
Minimum Known Magnitude (if available)			5	5
Alpha (if available)			0.000001	0.000001
Adversarial Method (if available)				FGS
eps (if available)				0.1 * abs(max(x_i))
Average Validation Confidence (AUC in Adversarial Case)	0.6987	0.6925	0.7069	0.7781
Known Validation Confidence (if available)	0.6987	0.3923	0.4158	
Epoch Taken	24934	65096	24638	10360

Table 4.2: Combination of parameters that reaches the highest evaluation scores for four models. The first three models are evaluated on the confidence of the validation set, and the last one relies on the AUC score. A blank cell means such a condition does not apply to this model.



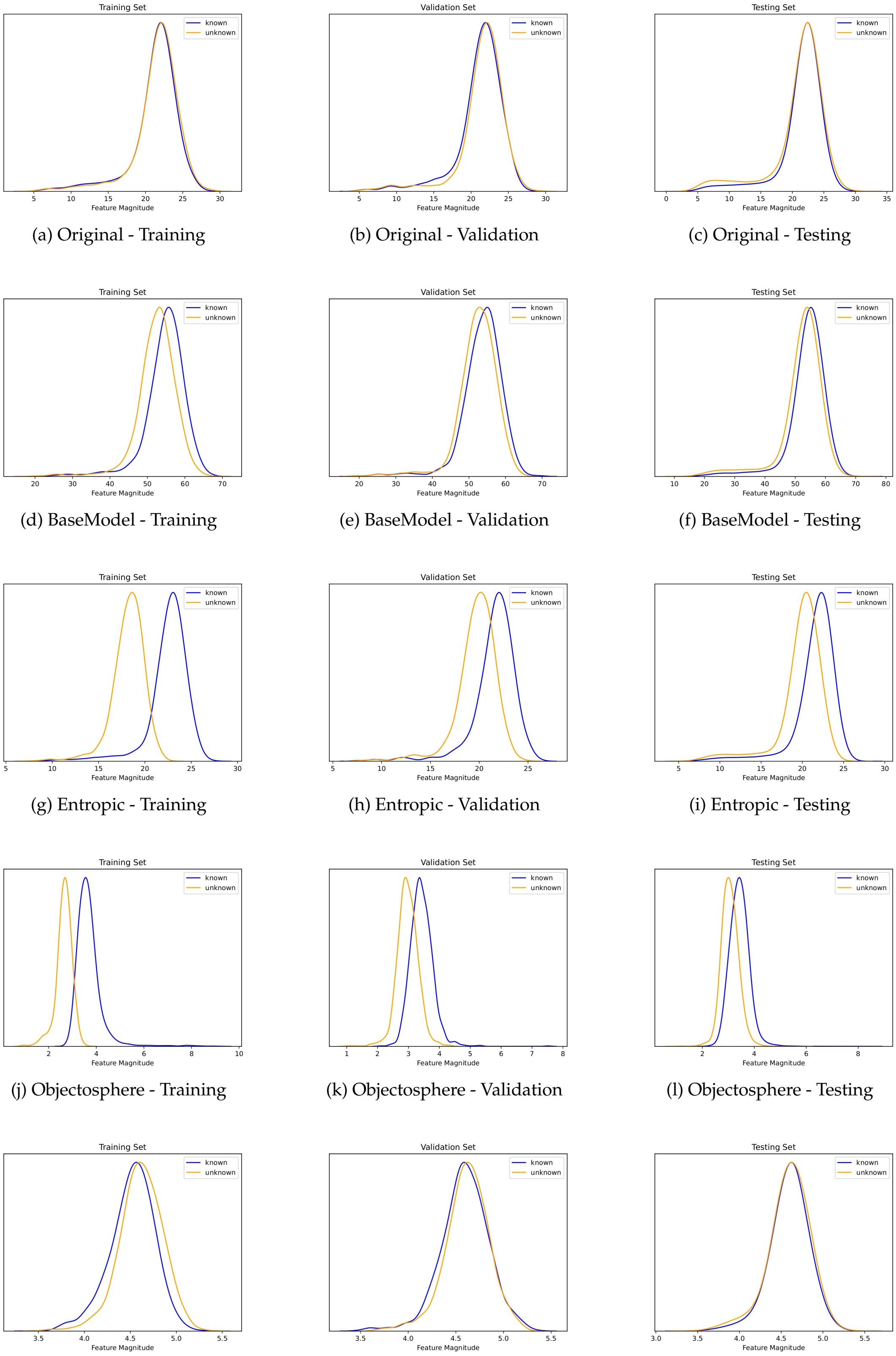
(a) Base Model with Softmax

(b) Known Unknowns $G2$ with Entropic

(c) Known Unknowns $G2$ with Objectosphere

(d) Adversarial samples with Objectosphere

Figure 4.2: Confidence plot for four shallow networks, with the parameters listed in Table 4.2. (a) Only knowns from $G1$ are involved in the training and validation of the base model. (b)-(c) Knowns from $G1$ and unknowns from $G2$ are involved in the training and validation. (d) Knowns from $G1$ and generated adversarial samples as unknowns are involved in the training. *AUC-Probability* and *AUC-Magnitude* are calculated as the evaluation metrics for validation set. *AUC-Probability* is the same as the AUC score calculated by the softmax values. *eps-coef* times $|\max(x)|$ equals to ϵ.



(a) Original - Training

(b) Original - Validation

(c) Original - Testing

(d) BaseModel - Training

(e) BaseModel - Validation

(f) BaseModel - Testing

(g) Entropic - Training

(h) Entropic - Validation

(i) Entropic - Testing

(j) Objectosphere - Training

(k) Objectosphere - Validation

(l) Objectosphere - Testing

(m) Adversarial - Training

(n) Adversarial - Validation

(o) Adversarial - Testing

Figure 4.3: Extracted feature magnitude for known and unknown samples. (a)-(c) are the original features. (d)-(f) are the features extracted from the base model, with the parameters listed in Table 4.2. (g)-(i) are from the model trained by $G2$ with entropic loss. (j)-(l) are from the model trained by $G2$ with objectosphere loss. (m)-(o) are from the model trained by adversarial samples with objectosphere loss and evaluated by the AUC scores for softmax value (*AUC-Probability* in Figure 4.2(d)).

Known Unknowns $G2$

Known unknowns from $G2$ are then involved in the model training and the loss function is replaced by the entropic open-set loss. Similarly to the base model, we adjust the parameters shown in Table 4.1 and find that the setup listed in Table 4.2 leads to the best average validation confidence (0.7). In Figure 4.2(b), the unknowns have a close to 1 confidence but only 0.4 for knowns. This is discussed in Chapter 5. The entropic open-set loss creates some separation in Figure 4.3(g)-4.3(i), i.e. pushes the unknown magnitude to below 23 and mainly falls in the range 15-20, while known magnitude has a similar range with the original sample features.

When applying the objectosphere loss, two more parameters should be considered, `MinimumKnownMagnitude` and α, as shown in Table 4.3. We choose to use 5 and $1e-6$, respectively. The effect of the former could be observed in the feature magnitude plots 4.3(j)-4.3(l), where the separation is similar to the entropic case, but the known magnitude is reduced to about 4 (not exactly 5) and about 3 for unknowns. So the objectosphere loss works in the way that we expect but does not push all knowns to above 5 and unknowns to 0. Further, in Table 4.2, our optimizer is changed to Adam and the normalization is applied to the inputs before forwarding to the first hidden layer. The validation confidence is close to the entropic case above.

Special Parameters	Choices
Minimum Known Magnitude	5, 50
α	0.01, 0.001, 0.0001, 0.000001
Generating adversarial technique	FGS, FGV
ϵ (Part I)	0.05, 0.1, 0.9, $\max((0.95)^{epoch}, 0.01)$
ϵ (Part II)	Absolute Maximum for each x or for all data

Table 4.3: Special parameters to choose in objectosphere loss and generating adversarial samples. The first two rows are the choices for objectosphere loss, and the last three rows are for generating the adversarial samples.

Adversarial Images

The known unknowns from $G2$ are replaced by the adversarial samples generated through FGS and/or FGV. Through the experiments, FGS works better than FGV and the latter is discussed in Chapter 5. The last column in Table 4.2 lists the parameters that achieve the best AUC score. The AUC score for comparing the softmax value (probability) and real label is used to evaluate the performance, while AUC for the feature magnitude method is not able to capture the improvement and overfitting of the network, which is discussed below as well. Normalization of inputs, optimizer Adam, and other parameters for objectosphere loss are the same as above. Though we consider the multiple versions of the first value in ϵ, constant value 0.1 is the most appropriate quantity.

Confidence is kept for the training set evaluation, where the confidence for unknowns is the confidence of classifying the adversarial samples as class -1, but the validation set implements AUC instead. AUC is computed for two scores against the real labels, softmax value and feature magnitude. We observe that time to train with those two values differ, i.e. epochs took to reach the maximum point for *AUC-Probability* and *AUC-Magnitude* are different in Figure 4.2(d), and when normalization is not applied before forwarding, magnitude AUC always requires more epochs, vice versa. This also results in a good separation for the training set but weakens the face recognition experiment. Thus with normalization and $\epsilon = 0.1 * |\max(x)|$, we rely on the AUC for softmax value and reach 0.7781. Unfortunately, the separation in Figure 4.3(m)-4.3(o) does not improve from the original and base model. We can find the proof in Figure 4.2(d). The AUC for magnitude reaches the maximum fast, then falls below 0.5, which indicates that the known magnitude and unknown magnitude cannot be differentiated.

4.4 Face Recognition Experiments

The above-trained models follow the ArcFace R100 network in the extractor to further detail the features as illustrated in Figure 4.1. Then the features extracted by two networks are forwarded to similarity score computation.

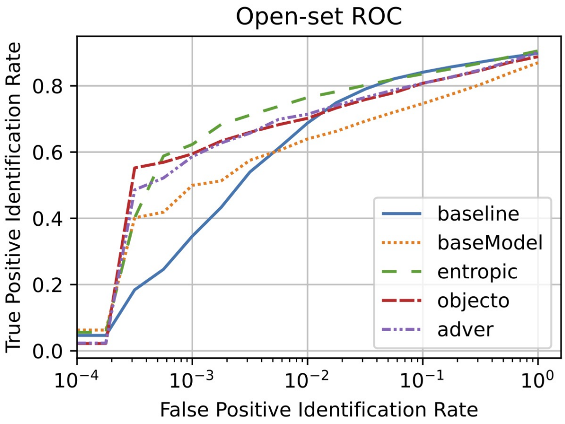
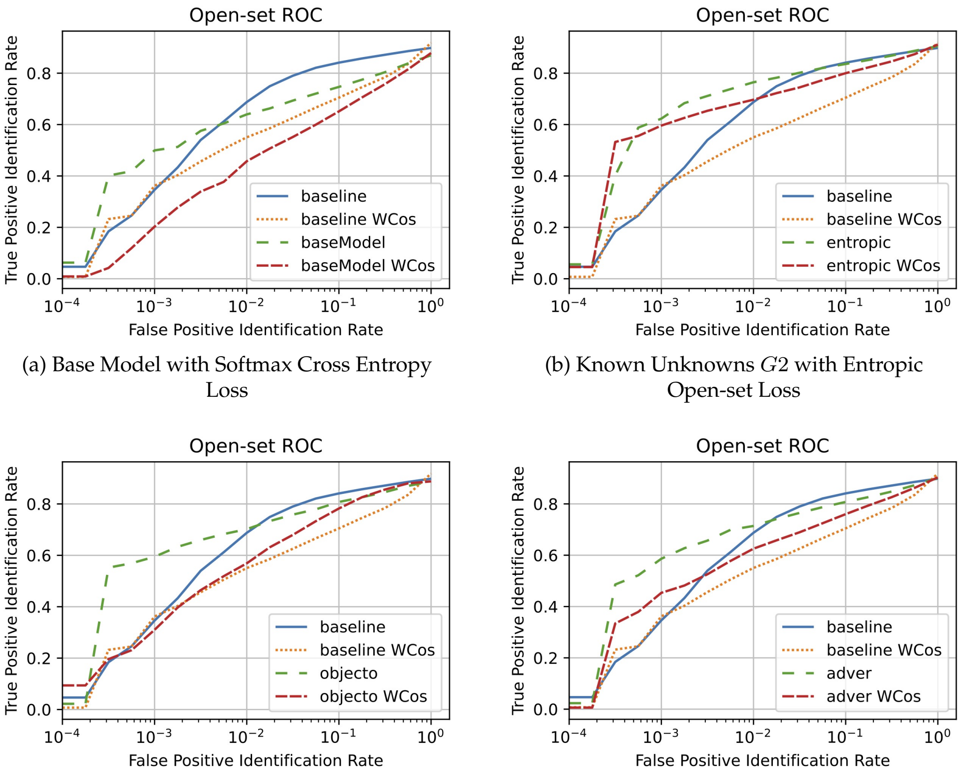


Figure 4.4: Open-set ROC plots for 5 models: *baseline* without the extra trained network; *baseModel* as the extra extractor trained only with known samples; *entropic* as the extra extractor trained by unknowns $G2$ and entropic open-set loss; *objecto* as the extra extractor trained by unknowns $G2$ and objectosphere loss; *adver* as the extra extractor trained by generated adversarial samples and objectosphere loss.

Figure 4.4 shows the open-set ROC curve composed of the cosine similarity score (Cos). The trained models, including the base model, have a higher TPIR value when the FPIR is below 0.01. The baseline outperforms most of them as FPIR increases. Base model and known unknowns $G2$ model with objectosphere loss end below baseline, and only unknown $G2$ model with entropic loss successively improves baseline thoroughly. Two models trained with objectosphere loss have a similar trend, are highly overlapping, and have a starting point lower than baseline. The first model *objecto* is trained with $G2$ as unknowns and the second model *adver* is trained with the generated adversarial samples as unknowns. But only $G2$ appears in the validation and testing, as well as in the FR experiments. So $G2$ is the known unknowns in the first case, but the unknown unknowns in the second case. Since they have similar performance in the FR experiments, our basic assumption that we can replace known unknowns $G2$ for training with adversarial samples works as expected. Also, our trained models bring some benefits to the baseline, and training with the unknown adversarial samples does not break up those benefits.

We suppose that the weighted cosine similarity (WCos) improves the performance when the feature magnitudes of probe samples could be better separated (Günther et al., 2020). In the last column of Figure 4.3, the feature magnitude plots for the testing set, the complete overlap is exhibited, except for a small moving of unknowns for two known unknowns $G2$ models. Thus, it is reasonable to assume that with the same FPIR, the weighted cosine similarity could not achieve a higher TPIR value than the unweighted case. Putting 10 curves in one plot makes it hard to



(a) Base Model with Softmax Cross Entropy Loss

(b) Known Unknowns $G2$ with Entropic Open-set Loss

(c) Known Unknowns $G2$ with Objectosphere Loss

(d) Adversarial Samples with Objectosphere Loss

Figure 4.5: Compare WCos and Cos with respect to the baseline case. Each plot contains four curves: baseline experiment with cosine similarity score and weighted cosine similarity; Cosine and weighted cosine similarity scores for experiments with a trained network as an extra extractor.

separate each line, thus we create the single comparison for WCos and Cos of each model as in Figure 4.5. Only the WCos score for the base model is lower than for baseline, the rest more or less have better WCos than baseline. In 4.5(a), both WCos curves are worse than Cos due to the full overlap of magnitude. 4.5(c) and 4.5(d) do not have similar WCos, though they are both trained by objectosphere loss and have close Cos's in Figure 4.4. This discrepancy is probably due to the existence of adversarial samples. Although the cosine distance is similar for features extracted from two models and probably the relative magnitude of deep features are also similar, the absolute magnitude might differ. Therefore, we get similar Cos curves but different WCos curves. In 4.5(b), WCos is closer but still worse than Cos, this also proves that the separation of the testing set is not well enough.

Chapter 5

Discussion

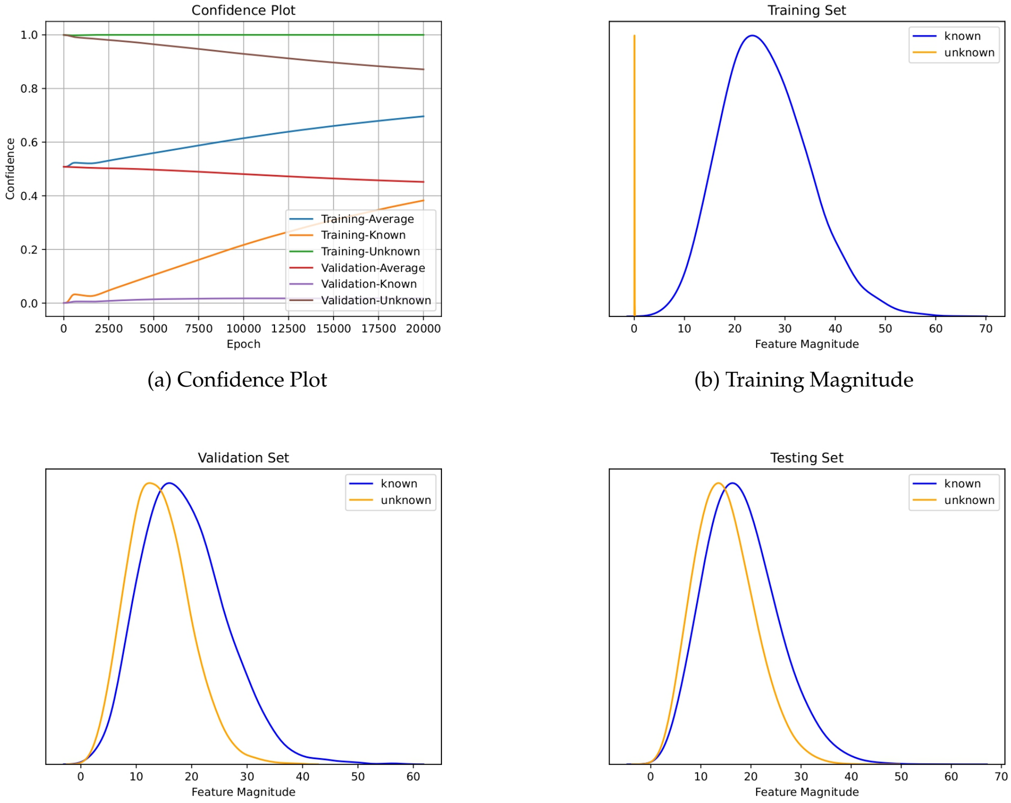
IJB-C Dataset Size

This thesis is built based on the paper 'Watchlist Adaptation: Protecting the Innocent' from Günther et al. (2020), where the UCCS dataset and VGG2 face recognition network are used instead. Their training set contains 11,299 of 11,315 knowns and 15,792 of 15,551 unknowns (and background as well) and is twice larger than the IJB-C galleries. They achieve a good separation on the magnitude of deep features from the testing set by a three-layer fully-connected network. The performance of the WCos curve is also better than that of the Cos curve. According to that paper, the network layer size is 128by64. In Figure 5.1, the same number of neurons provides under-fitting confidence and good separation in training magnitude but a weird large known magnitude.

The combinations 1024by512 and 2048by1024 perform better in the IJB-C case. Also, in the above-mentioned paper, the feature magnitude separation of probe samples is more powerful, i.e. unknowns are approaching 0 and knowns are more spread but less overlapping with unknowns. In our experiments, the overlapping in the probe set is always severe. We attribute this phenomenon to the different nature of the dataset, the resolution and face size are different, and IJB-C does not even introduce the background factor. It is also probable that some key parameters need to be fine-tuned to fit different datasets, which we do not figure out here.

Weighing the Training Samples

Except for the base model, we do not assign weights to the training samples. We do not expect the balancing in the base model will affect the results as apparent as in the other models, since none of the known classes can have a weight as large as the unknowns. When we use $G2$ as the known unknowns, then there are 6,011 samples with label -1, and 5,588 samples are distributed into 1,772 classes. That is, the number of unknown samples is much larger than the number of samples in a single known class. It is not surprising that the ability to identify the unknowns from knowns is better trained than that of classifying between each known class. This phenomenon can be found in the confidence plots for four models in Figure 4.2. The validation confidence for knowns can reach 0.7 when only knowns are involved in the network training. However, it is around 0.4 when unknowns join the training process in Figure 4.2(b) and 4.2(c), and both training and validation confidence for unknowns is approaching 1. It seems that the known samples are underrepresented which indicates that the weight for the unknown samples in the loss function is too high. Thus, we attribute the severe drop in the validation confidence to the heavy weights on the unknown samples, and adding the weight may also improve the performance the WCos curves. It is interesting to investigate the effects of weighing the known and unknown samples on the confidence plot, feature magnitude plot, and the WCos curve performance in future work.



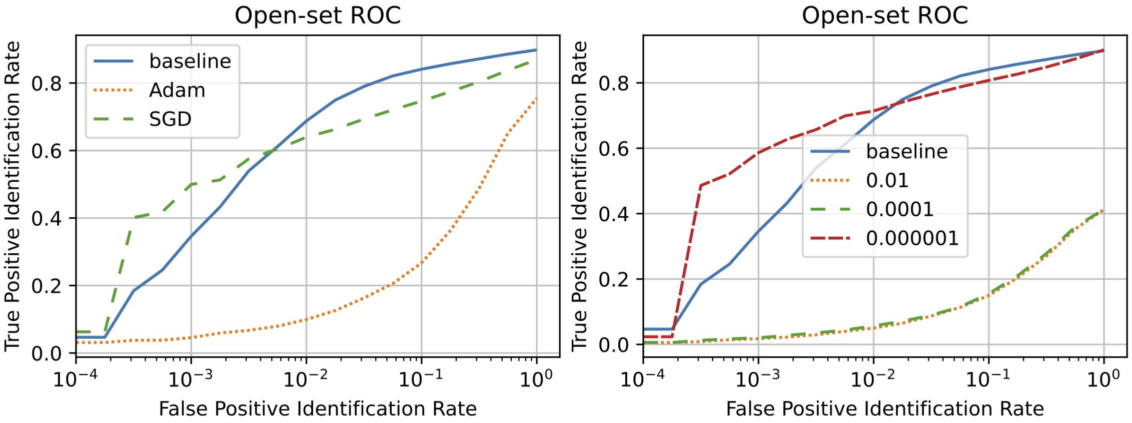
(a) Confidence Plot

(b) Training Magnitude

(c) Validation Magnitude

(d) Testing Magnitude

Figure 5.1: Known unknowns $G2$ model trained with objectosphere loss with 128by64 layer size. The other parameters are the same as the model in Figure 4.2(c).



(a) Base model with optimizer SGD and Adam

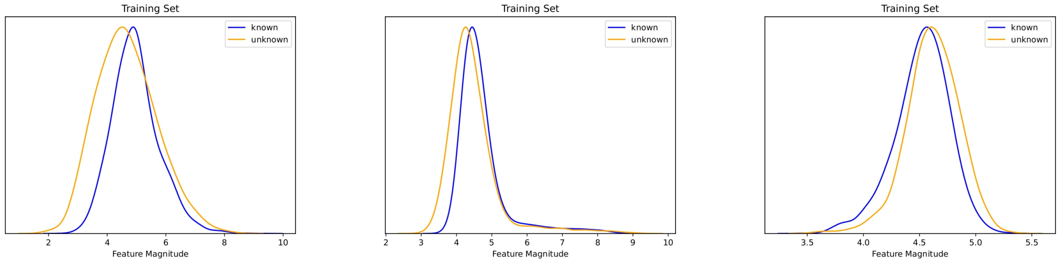
(b) Adversarial model different α

Figure 5.2: Open-set ROC plots for different optimizer or different α. (a) is the open-set ROC curve (Cos) for the baseline and base models with optimizer SGD and Adam. (b) is open-set ROC curve (Cos) for the baseline and models trained by adversarial samples with objectosphere loss with different α.

Optimizer and α effects on open-set ROC curve

One observation is regarding the optimizer. In the base model training, while using Adam with learning rate $1e-4$, the model reaches similar confidence with 10 times fewer epochs, but a horrible open-set ROC curve when joining the FR process, as shown in Figure 5.2(a), where the endpoint for Adam is 5% lower than SGD. The open-set ROC situation reverses in the known unknowns $G2$ with objectosphere loss model training. In both cases, confidence for Adam is higher than for SGD by 3% to 6%. The evaluation metrics (confidence) we use in model training have no apparent relationship with the evaluation for the face recognition experiment (Cosine Similarity) since the confidence uses the softmax value of prediction, and the cosine similarity computes the distance between features instead. This is further discussed below. The choice of optimizer should depend on the network structure as well as the dataset, though Adam usually takes less time to train as in Table 4.2.

Another observation happens in both the known unknowns $G2$ with the objectosphere loss model and the adversarial model. Normalization is applied to all the models trained here. Recall that the objectosphere loss (3.4) is the summation of the entropic loss and a weighted feature magnitude. Theoretically, the value of α should depend on the squared magnitude of the deep features and make sure the entropic loss is still dominating the entire loss function. In Figure 5.3, the adversarial model magnitude separation of the training set, decreasing α from 0.01 to 0.0001 and 0.000001 makes the separation more unobtrusive (though the separation is subtle even in 0.01 case), but increases the AUC scores for softmax values by 30% and 2%, respectively. Similarly, in the known unknown model, this change increases the validation confidence by $> 10\%$. But decreasing α has an obvious improvement in cosine similarity, where 0.01 and 0.0001 even could not reach the endpoint of the baseline as shown in Figure 5.2(b). The change in separation is expected since if we put more loss weight on the magnitude, then the model will learn the magnitude better. But the clip fall of the open-set ROC Curve is not, which is explained in the next paragraph.



(a) $\alpha = 0.01$

(b) $\alpha = 0.0001$

(c) $\alpha = 0.000001$

Figure 5.3: Confidence Plot for FGS adversarial samples with different feature magnitude α. Only the feature magnitudes for the samples in the training set are exhibited.

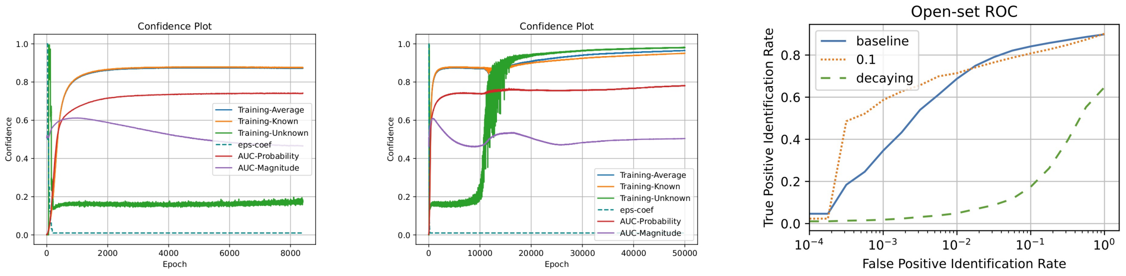
Magnitude Separation vs. open-set ROC Curve

We once wanted to build a proportional relationship to feature magnitude separation and open-set ROC curve performance. This is not exactly true even in Dhamija et al. (2018) though they obtained a good separation in the probe set. The second half of the objectosphere loss focuses on the magnitude of features, but not on the separation of feature space, which is required for calculating the cosine distance. We are not able to conclude our observation in the same way. As in Figure 4.3(i), 4.3(l), and 4.3(o), known unknowns $G2$ models with either entropic or objectosphere loss have similar but small separation, and the adversarial model is neglectable, but in Figure 4.4, the advantage of known unknowns does not completely overcome that of adversarial samples. This should again attribute to the non-separable problem that happens on dataset IJB-C because samples we trained with are images but we have frames in the probe/testing set. The quality of the probe set is supposed to influence the feature grabbing and recognition severely.

Thanks to Mr. Rafael Henrique Vareto, who figured out an implementation error on the entropic loss, and makes it possible to explain the bad open-set ROC Curve when the weight α is large. The entropic loss used throughout this thesis is taking the mean of loss for each sample, instead of the summation shown in equation (3.3), which makes the entropic loss smaller than we expect. This affects more on the models trained by the objectosphere loss since the squared magnitude with a large weight α is dominant the loss and thus the model focuses on learning the magnitude but not the feature space. Thus, a higher α could result in a good magnitude separation and bad open-set ROC Curve simultaneously. We draw our conclusions based on the old implementation since $\alpha = 0.000001$ balanced the loss, but the performance might be different from the new one and it is worth investigating further.

FGS with Decaying ϵ

As mentioned in the last chapter, decaying ϵ is applied to generate the adversarial samples. The real decaying happens on *eps-coef*, but ϵ decays with it as well. Since 0.95^{epoch} decays to 0.001 after 134 epochs, the changes to the inputs become too small before the network is trained well to identify them. We try to slow the decay rate to change once per 10 or 20 epochs, but none of them works well and results in the unstable unknown training confidence. The pattern becomes better when we set the rate to change per 30 epochs. We set up the early stopping criterion for 1,000 no improvement epochs on the validation metrics, and have the corresponding confidence plot in Figure 5.4(a). If we free that restriction and leave the training run for 50,000 epoch, we get Figure 5.4(b) and an improvement of AUC scores for softmax value (*AUC-Probability*) from

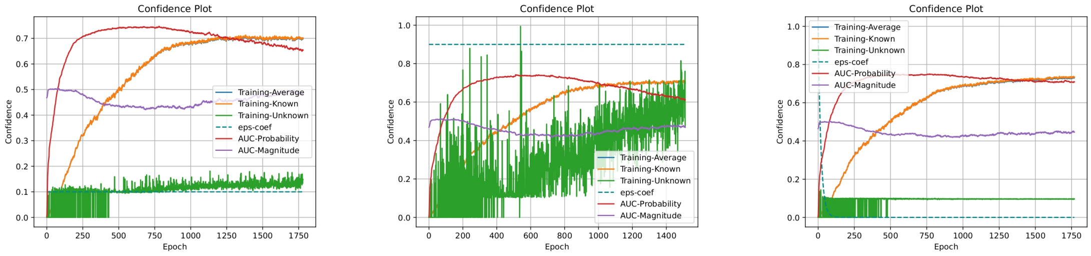


(a) Zoom in to first 10,000 epochs

(b) 50,000 epochs

(c) Open-set ROC Curves Comparison

Figure 5.4: Performance of using a decaying ϵ. (a) is the confidence plot for the first 10,000 epochs training. (b) elongates the training to 50,000 epochs. (c) includes the open-set ROC plots for baseline, fixed *eps-coef*, and decaying *eps-coef* / ϵ.



(a) eps-coef $= 0.1$

(b) eps-coef $= 0.9$

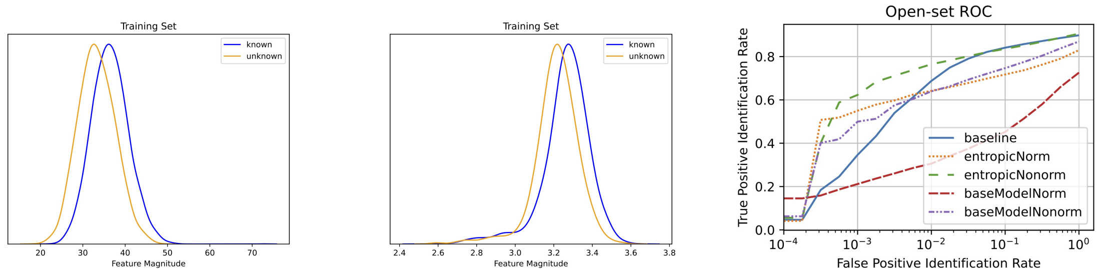
(c) Decaying eps-coef

Figure 5.5: Confidence Plot for adversarial samples generated by FGV. (a) has a *eps-coef* $= 0.1$, which is the same as the one that has the best performance in FGS above. (b) has *eps-coef* $= 0.9$, which is quite large. (c) has a decaying *eps-coef* / ϵ.

0.74 to about 0.8. The ϵ in this experiment has a lower base value 0.5. As we expect, given $\epsilon = max(0.5^{epoch//30}, 0.01)$, the initial adversarial samples are distinct from the knowns, so that easy to classify and the confidence jumps up suddenly. It is followed by a drastic drop because *eps-coef* falls to 0.01 fast and the difficulty to identify the unknowns is raised. Then, with the minimum *eps-coef*$= 0.01$, we zoom out the scale to 5.4(b). The model starts to learn the unknowns similar to Figure 4.2(d) and finally approaches 1 again. Though the model obtains a small increase in AUC than the *eps-coef* $= 0.1$ case, it gets a worse open-set ROC curve in 5.4(c). We expect a similar pattern will appear in the experiments with a higher base value like 0.99 since it will decay to 0.01 fast as well. Keeping slower the decay rate might be able to improve the performance.

FGV

FGV is tested for adversarial training. It takes less than 1,000 epochs to finish training for both AUCs. The training confidence for adversarial samples is highly fluctuating. Changing the first part of ϵ (*eps-coef*) as we discussed above does not help, shown in Figure 5.5. When *eps-coef* $= 0.1$, confidence fluctuates around 0.1 but with a slight uptrend. *eps-coef* $= 0.9$ results in random-noise-like confidence but with an increasing trend. Decaying *eps-coef* makes the confidence approximate to 0.1 after 500 epochs. In addition, when we train the model through the AUC scores calcu-



(a) Base model Normalization

(b) Known unknowns $G2$ model with entropic loss Normalization

(c) Open-set ROC for Normalization effects

Figure 5.6: Normalization of inputs for the base model and known unknowns $G2$ entropic model training. (a) is the deep feature magnitudes for the training set samples extracted by the base model training by the normalized inputs. (b) is by the known unknowns $G2$ model with entropic loss and normalized inputs. (c) is the open-set ROC curves for baseline, the base model with or without normalization, known unknowns $G2$ entropic model with or without normalization.

lated by the softmax value (*AUC-Probability*), the training feature magnitudes for knowns and unknowns are close to 5 but heavily overlapped, and close to 10 for the model training through the AUC calculated by the feature magnitudes (*AUC-Magnitude*). Considering those models are not well-trained before the validation AUC decreases, we conclude that FGV is not appropriate for the FR task with the IJB-C dataset.

Normalization on Features

Some of the experiments, known unknowns $G2$ and adversarial models with objectosphere loss, introduce the normalization of inputs in the network, i.e. normalization becomes the first layer and then followed by the first fully-connected layer. It shortens the training time and is expected to perform better. We also test the effects of normalization for the base model and the known unknowns $G2$ model with entropic loss, but the above conclusion and ideas for those two models are drawn without the normalization, and they might be different when normalization is involved in the training. When normalization is involved, it takes 60k epochs for the model trained with entropic loss and pushes the feature magnitude to a smaller range, 3.0-3.6, as in Figure 5.6(b). But it has a worse separation which looks like the situation happens on the training set features extracted by the base model without normalization as in Figure 4.3(d). The base model requires $> 450k$ epochs for training and results in a better separation and magnitude range of 30-40 as in Figure 5.6(a). Figure 5.6(c) exhibits that normalization is not helpful in the open-set ROC curve for those two models, especially the base model with normalization results in a drastic decrease in the TPIR performance. Therefore, the normalization does not bring benefits to these two models in training speed, feature magnitude separation, and the cosine similarity curve performance. There might be some related coefficients that require to be fine-tuned caused by the normalization. We still expect the normalization could make better separations and open-set ROC curves in the base model and known unknowns $G2$ entropic model.

Chapter 6

Conclusion

We deal with the open-set face recognition tasks with different loss functions and different sources of unknown samples. There are three situations: first, a model is only trained and validated with the samples that we care about and no unknown sample involved; second, a model is trained with both knowns and unknowns and the subjects appear in validation exist in the training set; third, a model is trained with knowns and the corresponding generated adversarial samples, i.e. as unknown samples, but the model has never seen the unknown subjects in the training. In a general face recognition task, we are supposed to extend the feature extraction step by adding the previous trained shallow network to the end of the ArcFace R100 network so that the resulting open-set ROC curve should perform better. The first one is trained by plain softmax loss, the second one experiences entropic open-set loss and objectosphere loss, and the third one uses objectosphere loss. We prove that all four situations have a positive effect on recognizing the deep feature of each face, especially for a small FPIR. When comparing the open-set ROC curves for the second case with objectosphere loss and the third case, we prove that known unknowns from $G2$ can be replaced by the adversarial samples without losing performance. Unfortunately, our work does not reflect the feature magnitude separation advantage of the entropic open-set loss and objectosphere loss. The separations are successful only for the training set, and the effects decrease when facing the probe set. The difference between the UCCS dataset and IJB-C may contribute to the magnitude problem. Also, objectosphere loss does not successfully differentiate the known samples and adversarial samples generated from them. In addition to what we mentioned in the discussion, it is worth applying some other adversarial generating techniques, looking for the possible relationship between parameters of FGS and FGV and the dataset, and using different networks for base feature extraction.

Appendix A

Attachements

List of Figures

List of Tables

Bibliography

Anjos, A., El-Shafey, L., Wallace, R., Günther, M., McCool, C., and Marcel, S. (2012). Bob: a free signal processing and machine learning toolbox for researchers. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1449–1452.

Bansal, A., Nanduri, A., Castillo, C. D., Ranjan, R., and Chellappa, R. (2017). UMDFaces: An Annotated Face Dataset for Training Deep Networks. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 464–473.

Bendale, A. and Boult, T. E. (2016). Towards Open Set Deep Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572.

Beveridge, J. R., Phillips, P. J., Bolme, D. S., Draper, B. A., Givens, G. H., Lui, Y. M., Teli, M. N., Zhang, H., Scruggs, W. T., Bowyer, K. W., Flynn, P. J., and Cheng, S. (2013). The Challenge of Face Recognition from Digital Point-and-Shoot Cameras. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8.

Beveridge, J. R., Zhang, H., Draper, B. A., Flynn, P. J., Feng, Z., Huber, P., Kittler, J., Huang, Z., Li, S., Li, Y., Kan, M., Wang, R., Shan, S., Chen, X., Li, H., Hua, G., Štruc, V., Križaj, J., Ding, C., Tao, D., and Phillips, P. J. (2015). Report on the FG 2015 Video Person Recognition Evaluation. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–8.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. (2018). VGGFace2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74.

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., and Ma, Y. (2015). PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032.

Chen, J.-C., Patel, V. M., and Chellappa, R. (2016). Unconstrained face verification using deep CNN features. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9.

Chowdhury, A. R., Lin, T.-Y., Maji, S., and Learned-Miller, E. (2016). One-to-many face recognition with bilinear CNNs. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. (2017). EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE.

de O. Cardoso, D., Gama, J., and França, F. M. G. (2017). Weightless neural networks for open set recognition. *Machine Learning*, 106(9):1547–1567.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699.

Dhamija, A. R., Günther, M., and Boult, T. E. (2018). Reducing Network Agnostophobia. In *Advances in Neural Information Processing Systems*, volume 31, pages 9157–9168.

Ding, G. W., Wang, L., and Jin, X. (2019). AdverTorch v0.1: An Adversarial Robustness Toolbox based on PyTorch. *arXiv preprint arXiv:1902.07623*.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. *International journal of computer vision*, 88(2):303–338.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. In *ICLR 2015 : International Conference on Learning Representations 2015*.

Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. (2016). MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In *European Conference on Computer Vision*, pages 87–102.

Günther, M., Cruz, S., Rudd, E. M., and Boult, T. E. (2017a). Toward Open-Set Face Recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 573–582.

Günther, M., Dhamija, A. R., and Boult, T. E. (2020). Watchlist Adaptation: Protecting the Innocent. In *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 21–32.

Günther, M., Hu, P., Herrmann, C., Chan, C. H., Jiang, M., Yang, S., Dhamija, A. R., Ramanan, D., Beyerer, J., Kittler, J., Jazaery, M. A., Nouyed, M. I., Guo, G., Stankiewicz, C., and Boult, T. E. (2017b). Unconstrained Face Detection and Open-Set Face Recognition Challenge. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 697–706.

Günther, M., Wallace, R., and Marcel, S. (2012). An open source framework for standardized comparisons of face recognition algorithms. In *ECCV'12 Proceedings of the 12th international conference on Computer Vision - Volume Part III*, pages 547–556.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Herrmann, C., Willersinn, D., and Beyerer, J. (2016a). Low-Quality Video Face Recognition with Deep Networks and Polygonal Chain Distance. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7.

Herrmann, C., Willersinn, D., and Beyerer, J. (2016b). Low-resolution Convolutional Neural Networks for video face recognition. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 221–227.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst.

Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. (2016). The MegaFace Benchmark: 1 Million Faces for Recognition at Scale. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4873–4882.

Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., Burge, M., and Jain, A. K. (2015). Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1939.

LeCun, Y. (1998). The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Levi, G. and Hassner, T. (2015). Age and Gender Classification using Convolutional Neural Networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 34–42.

Lin, T.-Y., RoyChowdhury, A., and Maji, S. (2015). Bilinear CNN models for fine-grained visual recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1457.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In *14th European Conference on Computer Vision, ECCV 2016*, pages 21–37.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). SphereFace: Deep Hypersphere Embedding for Face Recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746.

Maze, B., Adams, J., Duncan, J. A., Kalka, N., Miller, T., Otto, C., Jain, A. K., Niggel, W. T., Anderson, J., Cheney, J., and Grother, P. (2018). IARPA Janus Benchmark - C: Face Dataset and Protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165.

O'Toole, A. J., Castillo, C. D., Parde, C. J., Hill, M. Q., and Chellappa, R. (2018). Face Space Representations in Deep Convolutional Neural Networks. *Trends in Cognitive Sciences*, 22(9):794–809.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep Face Recognition. In *British Machine Vision Conference 2015*.

Phillips, P. J., Grother, P., and Micheals, R. (2011). Evaluation methods in face recognition. In *Handbook of face recognition*, pages 551–574. Springer.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Rozsa, A., Rudd, E. M., and Boult, T. E. (2016). Adversarial Diversity and Hard Positive Generation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 410–417.

Rudd, E. M., Jain, L. P., Scheirer, W. J., and Boult, T. E. (2018). The Extreme Value Machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):762–768.

Sáez-Trigueros, D., Meng, L., and Hartnett, M. (2018). Face Recognition: From Traditional to Deep Learning Methods. *arXiv preprint arXiv:1811.00116*.

Sankaranarayanan, S., Alavi, A., Castillo, C. D., and Chellappa, R. (2016). Triplet Probabilistic Embedding for Face Verification and Clustering. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8.

Sapkota, A. and Boult, T. E. (2013). Large scale unconstrained open set face database. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8.

Schnyder, J. (2021). Deep Adversarial Training for Teaching Networks to Reject Unknown Inputs.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Sun, Y., Wang, X., and Tang, X. (2014). Deep Learning Face Representation by Joint Identification-Verification. In *Advances in Neural Information Processing Systems 27*, volume 27, pages 1988–1996.

Sun, Y., Wang, X., and Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2892–2900.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. (2018). CosFace: Large Margin Cosine Loss for Deep Face Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A Discriminative Feature Learning Approach for Deep Face Recognition. In *European Conference on Computer Vision*, pages 499–515.

Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A. K., Duncan, J. A., Allen, K., Cheney, J., and Grother, P. (2017). IARPA Janus Benchmark-B Face Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 592–600.

Wolf, L., Hassner, T., and Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534. IEEE.

Yang, S., Luo, P., Loy, C. C., and Tang, X. (2016). WIDER FACE: A Face Detection Benchmark. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5525–5533.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Learning Face Representation from Scratch. *arXiv preprint arXiv:1411.7923*, (11).

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.