Masked Face Recognition

Vibhaakar Sharma (vibhs@stanford.edu)
Swathi Gangaraju (swathig1@stanford.edu)
Vishal K. Sharma (vks@stanford.edu)

Abstract

This work aims to build Masked Face Recognition model using existing Face Recognition algorithms and public masked face datasets. We used Labeled Faces in the Wild and Simulated Masked Face Recognition Datasets. We implemented transfer learning to retrain FaceNet model with Inception ResNet v1 and ResNet50 architectures and achieved <99.98% accuracy on the training set. We performed hyperparameter tuning to address overfitting on validation sets. We encountered many issues with generalizing the model to validation set and addressed some of the issues.

Problem and Motivation

We are amidst an ongoing pandemic. Face masks are recommended to control the spread of COVID-19. Face recognition is a popular mode of authentication which is now broken due to faces being covered by face masks. People are seen removing their face masks to authenticate, especially on their phones, risking public health. Face masks are now an added challenge to face recognition systems along with the variations in imaging conditions. Multiple prominent facial features like nose, mouth, and chin are covered with a mask which otherwise contributes significantly to the face recognition process. Our project aims to build Masked Face Recognition model using existing Face Recognition algorithms and public masked face datasets. The desired outcome is to recognize a masked face image.

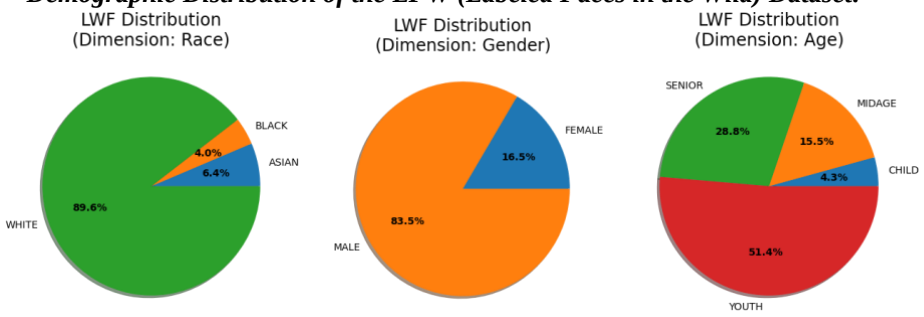
Related Work

Face Recognition is a computer vision task that has been extensively studied for several decades. Some of the notable works in solving Face Recognition problem are DeepFace proposed by Taigman et al.[23], FaceNet by Schroff et al.[15], BAIDU by Liu et al.[24] and VGGFace by Parkhi et al.[26] Our literature study showed that work was also done on occluded Face Recognition, specifically Masked Face recognition problem[3, 5, 7, 10] even before COVID pandemic started. However, the efforts have multiplied manifold since the start of pandemic [1,2,4,9]. Some of the approaches explored so far to solve masked face recognition problem include feature extraction to train model only on the un-occluded part of the face as noted Efficient Masked Face Recognition Method during the COVID-19 Pandemic paper by Walid Hariri[1], creation of Masked dataset using 'MasktheFace' application in the Masked Face Recognition for Secure Authentication paper by Aqeel Anwar, Arijit Raychowdhury[2] and GAN (Generative Adversarial Networks) based approach to unmask the masked face in the A novel GAN-based network for unmasking of masked face paper by Nizam Ud Din et al.[9] Most of these approaches propose complex model building or application creation from scratch to solve the problem. In the literature study we noticed that there are few public datasets available that are well-suited for Face Recognition research like Labeled Faces in the Wild dataset created by UMass[13], CASIA WebFace, MS Celeb 1M, VGGFace2 datasets[7]. There are fewer datasets available for masked faces. Zhongyuan Wang et al.[6] created Real-World masked face dataset and Simulated Masked Face Recognition dataset in 2020. While lot of research is underway to make Face Recognition systems more robust, there is also increasing concerns over Face Recognition programs creating privacy, security, accuracy, bias, and freedom issues[14]

Dataset

We used two public datasets. First dataset is the deep-funneled LFW (Labeled Faces in the Wild) dataset created by Gary B. Huang et al.[11]. This dataset has ~13,000 aligned and labeled images of faces collected from the web. The images are organized into 5749 folders with each folder corresponding to a person and their original dimensions are 250 x 250 pixels. This deep-funneled dataset is produced by rotating and resizing images to ensure that faces are at the center of the image. Second dataset used is Simulated Masked Face Recognition Dataset (SMFRD)[6]. This dataset has been created by a simulation mask-wearing application over the LFW dataset. We preprocessed the images from both datasets to have shape of 160x160x3 and 224x224x3 before running them through the models. This effort resulted in balanced dataset having total of 25,035 masked and unmasked images. While dataset has representation from different races, genders, and age groups, majority of the images belong to male, white and youth population. Below pie charts show the demographic distribution of the LFW (Labeled Faces in the Wild) dataset. We assumed this skewed distribution can affect model performance on certain minority population images and made sure we checked for gender, race, and age specific accuracies to analyze the impact and identify right action to address it.

Demographic Distribution of the LFW (Labeled Faces in the Wild) Dataset:



We selected folders (one per person) with 3 or more images and moved one of the images to validation set and another one to test set. For people with only two images in the dataset, we moved one of the images to either test or validation set. All remaining ones in the original dataset contributed to the training dataset. We achieved following distribution for Train/Validation/Test datasets:

Dataset	Train	Validation	Test	Split%(Train/Val/Test)
Masked Dataset	10559	1664	887	80.53/12.69/6.77
Unmasked Dataset	11449	1679	900	81.5/11.99/6.42
Total	**21988**	**3343**	**1787**	**80.08/12.33/6.59**

We also created a subset of 60 masked and 60 unmasked images by hand-picking higher resolution images from the above dataset. This subset avoids any obstructions to faces other than masks to aid with error analysis. We cropped the images to remove the noise in the background and just have 160x160x3 images where the face is centered.

Model Architecture

Figure 1: **FaceNet Model Structure**

FaceNet Architecture: We based our approach on the FaceNet model which is a face recognition system described by Florian Schroff, et al[15]. The model architecture is shown in Figure 1. The driver for us selecting this approach is that on LFW (Labeled Faces in the Wild) dataset, this system is reported to have an accuracy of 99.63%[15] on unmasked faces. Our hypothesis was if we use transfer learning technique on FaceNet and re-train the model with masked and unmasked faces, we should be able to achieve high accuracy given the proven architecture for Face Recognition for facial feature extraction. FaceNet system extracts high-quality features from the face image and generates a 128-element vector (a.k.a. face embedding) representation of these features.

Figure 2: **Triplet Loss**

Facenet model is a deep convolutional neural network optimized via a triplet loss function that produces the embeddings of a same person images with a closer Euclidean distance than embeddings of different people images as shown in Figure 2.

Triplet loss function:

$$\sum_{i}^{N}[||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha$$

where
$f(x_i^a)$ is an embedding vector of anchor input
$f(x_i^p)$ is an embedding vector of positive input of the same class as anchor
$f(x_i^n)$ is an embedding vector of negative input of a different class from anchor
α is a margin between positive and negative pairs
N is number of examples.

ResNet50 Architecture: We also used ResNet50 as an alternate to Inception-Res v1 in deep architecture. ResNet[27], short for Residual Networks is a classic neural network used as a backbone for many computer visions tasks. The fundamental breakthrough with ResNet was that it allowed us to retrain extremely deep neural networks with 150+layers successfully by reformulating the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. The ResNet-50 model consists of 5 stages each with a convolution and identity block. Each convolution block has 3 convolution layers, and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters. This model has 3.8 billion FLOPs.

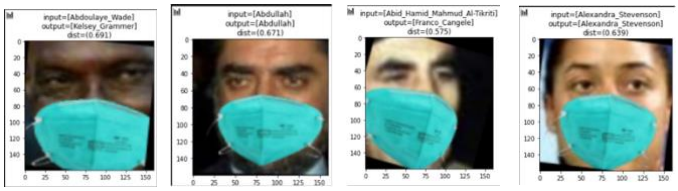
Method and Approach

Our project involved two main steps: 1) Retrain the existing Facenet models with weights for our dataset of unmasked and masked images optimizing for triplet loss. 2) Produce embeddings from retrained model for Train/Validation/Test sets and calculate accuracy using K-Nearest Neighbors (KNN) as classifier for face recognition. We ran the FaceNet model with the pre-trained weights to produce

embeddings to establish our baseline accuracy. We experimented with two deep architectures (Inception-ResNetv1 and ResNet50) and calculated the accuracy and FAR (False Acceptance Rate) for Train, Validation and Test datasets.

Andre Agassi *Annette Lu* *Abdullah Wade* *Dustin Hoffman* *Alexandra Steveson*

Sample Input: 

Sample Expected Output: 

Experiments and Hyper Parameter Tuning

Experiment Metrics: Based on our research of Face Recognition models, we selected Accuracy and False Acceptance Rate as evaluation metrics. False Acceptance Rate is significant in situations where Face Recognition is used for identity authentication.

Accuracy: (TP+TN)/(TP+FP+TN+FN); False Acceptance Ratio: FP/(TP+FP+TN+FN) where TP: True Positives; FP: False Positives; TN: True Negatives; FN: False Negatives

Experiment 1 – Using pre-trained FaceNet architecture and weights to set our baseline metrics: We used the pre-trained weights and the Keras FaceNet model provided by Hiroki Taniai[20]. This model has 22,808,144 parameters, accepts images with 160x160x3 size and pretrained on MS-Celeb-1M dataset. We first generated embeddings for our Train/Validation/Test sets running this model. Then we used KNN as the classifier over embeddings where output label is name of the person. We applied Ball Tree algorithm[21] to speed up the nearest neighbor search queries, in which the goal is to find the k points in the tree that are closest to a given test point by distance metric (Euclidean distance in our case). This approach helped us to establish a baseline to compare against the retrained models with our dataset (Transfer Learning) in the next set of experiments.

Experiment 2 – FaceNet architecture tuned with masked and unmasked dataset (Transfer Learning): We used pre-trained model and weights mentioned in Experiment 1. We optimized for Triplet loss as mentioned in FaceNet paper and performed following hyperparameter tunings:

Hyperparameter	**Value1**	**Value2**	**Value3**	**Finalized**
Learning Rate	0.1	0.05	0.01	0.01
Margin in Triplet loss	2	1.5	1	0.5
Number of layers retrained	Last 50% + added Dense & BatchNorm	Last 10 + added Dense & BatchNorm	Last one + added Dense & BatchNorm	Last one + added Dense & BatchNorm

We experimented with freezing different number of layers for transfer learning from retraining 50% layers to retraining only last layer. Based on the literature on transfer learning we also tried freezing all but batch norm layers[25] of the original model. We added a Dense and BatchNorm layer to retrain the model. We encountered vanishing gradient problem without having a BatchNorm layer with Dense.

Triplets' selection: We tried different approaches to select triplets (A, P, N) for retraining i.e., Anchor (A), Positive (P) same as Anchor and Negative (N) different person image. 1) Fixed triplets for full retraining which caused optimization over the set of triplets. 2) Provide a new random Negative to triplets after each epoch and 3) Provide a closest distance Negative after each epoch. Our approach (1) didn't generalize well and (3) was not optimal to compute embedding and Euclidean distances after each epoch. We retrained with approach (2) which was most optimal and generalizing one. We also created Triplets with 1) all three (A, P and N) from unmasked dataset 2) all three (A, P and N) from masked dataset and 3) A from unmasked dataset while P and N are from masked dataset. This allowed us to compare how our model is fitting to masked, unmasked, and combined datasets.

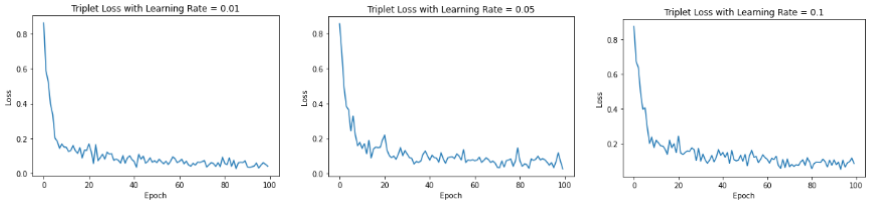
We initially used 25 epochs to get quick results and later increased the number of epochs to 75 and 100 to retrain the model longer. It took us ~6 hours to train the model with epoch = 75 and batch size = 32.

Experiment 3 - ResNet50 Architecture tuned with masked and unmasked dataset (Transfer Learning): We used ResNet-50 as the Deep Architecture in the FaceNet model to compare how this model would perform as a deeper architecture. This model has 50 layers, with 23,850,496 parameters and accepts 224x224x3 image size. We used pre-trained weights generated by running the model on ImageNet dataset. We used same hypermeters and Transfer Learning approach as in Experiment 2 and performed all the retraining.

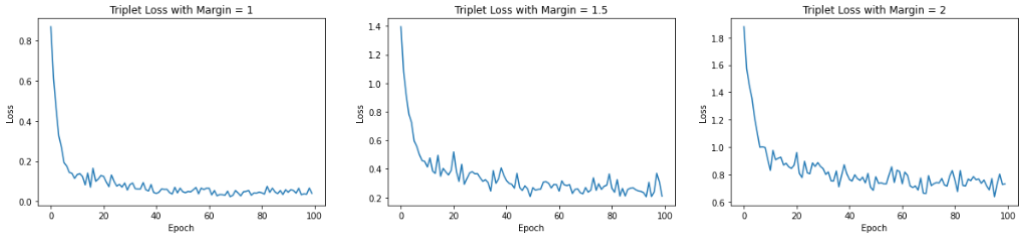
Experiment 4 – FaceNet architecture tuned with Curated Dataset: We manually curated a subset of 60 masked and 60 unmasked images. We repeated the Experiment 2 with FaceNet model to test our hypothesis that retraining with clear images without any noise like blur, occluded with mike or mug, multiple faces etc. may result in better accuracies. We observe a significant improvement in recognizing masked images against unmasked dataset.

Experiments Results

Triplet Losses with Different Learning Rates using FaceNet model for 100 epochs:



Triplet Losses with Different Triplet loss Margin (Alpha) using FaceNet model for 100 epochs:



Accuracy in % with triplet loss margin=0.5 and learning rate = 0.01:

Experiments	Train - Masked	Train - Unmasked	Validation- Masked	Validation - Unmasked	Test - Masked	Test - Unmasked
Experiment 1-pretrained FaceNet	99.99	100	53.89	14.67	50.28	12.78
Experiment 2-finetuned FaceNet	100.00	100.00	0.42	46.52	0.9	0.11
Experiment 3-finetuned ResNet50	99.98	100.00	0	46.52	0.11	0

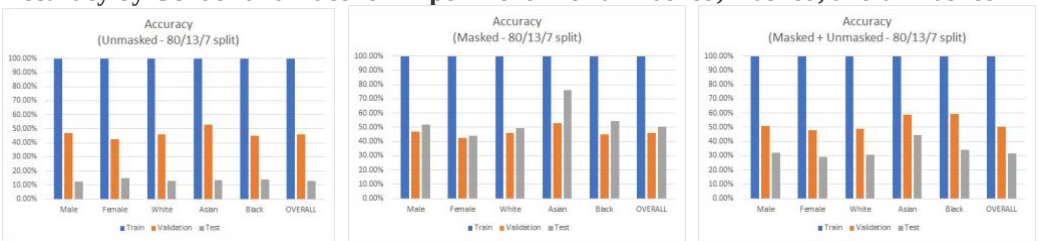
False Acceptance Rate in % with triplet loss margin=0.5 and learning rate = 0.01:

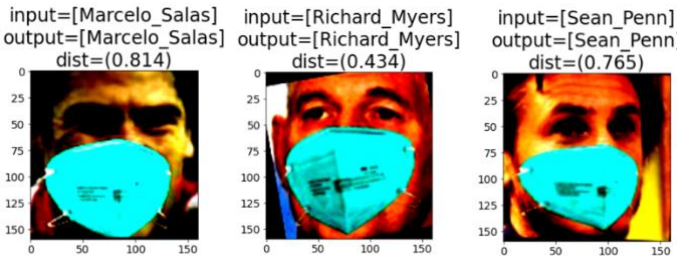
Experiments	Train - Masked	Train - Unmasked	Validation- Masked	Validation - Unmasked	Test - Masked	Test - Unmasked
Experiment 1-pretrained FaceNet	0.01	0	38.78	82.89	41.15	84.33
Experiment 2-finetuned FaceNet	0.02	0	99.58	53.48	99.10	99.89
Experiment 3-finetuned ResNet50	0.01	0	100	53.48	99.89	100.00

Accuracy in % for curated dataset of 60 unmasked images as anchors and 60 masked images as positive and negative:

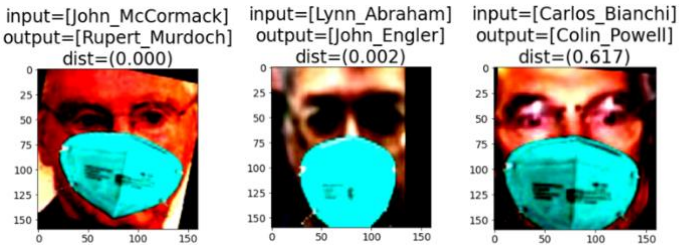
Dataset	Accuracy	False Acceptance Rate
Train (60 unmasked and 60 masked processed images)	86%	14%
Validation (16 masked images)	75%	25%
Test (3 masked images)	66%	34%

Accuracy by Gender and Race for Experiment 1 for unmasked, masked, and unmasked +masked datasets:





Correctly Recognized Images



Incorrectly Recognized Images

Analysis

Overall, we achieved < 99.98% accuracy over our training set. In Experiment 1, the FaceNet model with pre-trained weights performed better for masked images than unmasked images in validation and test sets. This is expected as we are using pre-trained weights obtained from FaceNet model trained on masked dataset. We did not observe significant performance difference between Inception-ResNetv1 vs ResNet50. We noted that retraining converged with best triplet loss with freezing all but last one layer + added Dense and BatchNorm layers, learning rate = 0.01 and margin equal = 0.5. Our retrained model generalized better with smaller margin of 0.5 which makes sense as large margin would force network to detect higher distances between positive and negative pairs. We noticed that in some cases positive pairs were more apart than the negative pairs causing high variance. On freezing different number of layers in deep architecture, we got the best results with freezing all but the last layer because our pretrained models were optimized on large datasets. During error analysis, we found that background noise, inferior clarity of the image and non-face mask obstructions are adversely affecting performance of the model. We conducted Experiment 4 with manually curated images and observed significant improvement in the validation and test accuracies. We revised our initial dataset split from 90%:5%:5% to 81%:12%:7% and observed marginal improvement in the accuracy rates.

We also observed that selecting right triplets for retraining is challenging. Our majority of dataset has 1 or 2 images per person limiting our options to select positive pairs. However, with 20K images in train dataset to consider every possible negative triplet option is computationally exhaustive. We optimized selecting different random negative pairs per epoch. We also observed vanishing gradients without the last BatchNorm layer block in the model during transfer learning as it is crucial to normalize the parameters without which huge distances between positive and negative pairs can pose vanishing gradients problem.

Contrary to our assumption based on race wise split of images in the dataset, we observed that accuracy on Asian faces was higher in experiments including masked, unmasked, combined dataset and accuracies of images attributed to Asian and Black races were higher than white race for combined dataset. This needs to be further analyzed to make conclusive statement. We also found accuracies on male images were better than images of female face. This can be explained by proportionally higher number of male images in training.

Conclusion and Next Steps

In this project we were able to apply Transfer Learning, FaceNet models and data processing skills to achieve <99.98% accuracy in the training dataset. We were able to solve the overfitting problem in the time available to achieve desired accuracy rate and False Acceptance Rate in the validation and test sets. We learnt a lot through our literature study, datasets, and error analysis. However significant work remains to be done to make this model application ready. Future work may include 1) Data augmentation to select more images per person. 2) Find ways to remove background noise in the image and remove images with other occlusions or obstructions from the dataset. 3) We did not come across a public masked face dataset that has balanced split with respect to gender, race, and age. Creating a balanced masked face dataset would be worthy effort to generalize the model for minority populations. 4) Triplet selection can significantly affect performance of the model so will try more efficient ideas for selecting triplets to retrain the model. 5) On the model front, further lowering the triplet loss margin and retraining with optimal set of triplets to improve the performance.

Contributions

Vishal K Sharma, Swathi Gangaraju and Vibhaakar Sharma collaborated closely and worked together on most of the tasks from ideation to execution on the project. Vishal primarily focused on Experiment 1, dataset analysis of diversity attributes, and generating visuals for all the experiments. Vibhaakar focused on AWS and GitHub setup, Experiment 2, 3 and 4, hyperparameter tuning and retraining. Swathi focused on literature research, data collection, data preparation, analysis, and report.

Acknowledgement

We are thankful to Fenglu Hong for her guidance and feedback on this project. Her inputs helped us overcome many challenges we encountered on this project.

References

1. Walid Hariri. “Efficient Masked Face Recognition Method during the COVID-19 Pandemic” In: *DOI: 10.21203/rs.3.rs-39289/v1 July 2020*
2. Aqeel Anwar, Arijit Raychowdhury. “Masked Face Recognition for Secure Authentication” In: *arXiv:2008.11104 25 Aug 2020*
3. Shiming Ge, Jia Li, Qiting Ye, Zhao Luo. “Detecting Masked Faces in the Wild with LLE-CNNs" In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
4. Rucha Golwalkar and Ninad Mehendale. “Masked Face Recognition Using Deep Metric Learning and FaceMaskNet-21" In: *SSRN: https://ssrn.com/abstract=3731223 November 16, 2020*
5. Md. Sabbir Ejaz, Md. Rabiul Islam, Md Sifatullah, Ananya Sarker “Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition” In: *International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) DOI: 10.1109/ICASERT.2019.8934543 May 2019*
6. Zhongyuan Wang et al. “Masked Face Recognition Dataset and Application” In: *arXiv:2003.09093 [cs.CV] 2020*
7. Insaf Adjabi, Abdeldjalil Ouahabi, Amir Benzaoui, Abdelmalik Taleb-Ahmed. “Past, Present, and Future of Face Recognition: A Review.” In: *Electronics, Volume 9, Year 2020, #8, Article #1188, ISSN: 2079-9292*
8. Nikolaos Passalis and Anastasios Tefas. “Learning Bag-of-Features Pooling for Deep Convolutional Neural Networks” In: *arxiv.org:1707.08105 Jul 2017.*
9. Nizam Ud Din, Kamran Javed, Seho Bae, and Juneho Yi. “A novel gan-based network for unmasking of masked face.” In: *IEEE Access, 8:44276–44287, 2020.*
10. Lingxue Song, Dihong Gong, Zhifeng Li, Changsong Liu, and Wei Liu. “Occlusion robust face recognition based on mask learning with pairwise differential Siamese network.” In: *Proceedings of the IEEE International Conference on Computer Vision, pages 773–782, 2019*
11. Gary B. Huang, Marwan A. Mattar, Honglak Lee, Erik Learned-Miller. “Learning to Align from Scratch” In: *NIPS 2012*
12. https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset
13. http://vis-www.cs.umass.edu/lfw/
14. Yi Zeng, Enmeng Lu, Yinqian Sun, Ruochen Tian. “Responsible Facial Recognition and Beyond” arxiv:1909.12935 [cs.CV] 2019
15. Florian Schroff, Dmitry Kalenichenko, James Philbin. “FaceNet: A Unified Embedding for Face Recognition and Clustering.” arxiv: 1503.03832.v3 [cs.CV] 2015
16. https://github.com/amitrani6/facial-recognition-system
17. https://www.kaggle.com/jessicali9530/lfw-dataset
18. https://www.kaggle.com/muhammeddalkran/lfw-simulated-masked-face-dataset
19. Christian Szegedy. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning” arxiv: 1602.07261 Aug 2016
20. Google Drive: https://drive.google.com/drive/folders/1pwQ3H4aJ8a6yyJHZkTwtjcL4wYWQb7bn
21. http://people.ee.duke.edu/~lcarin/liu06a.pdf
22. Ji, Qingge & Huang, Jie & He, Wenjie & Sun, Yankui. “Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. Algorithms.” 12. 51. 10.3390/a12030051.
23. Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.
24. Liu, J.; Deng, Y.; Bai, T.; Huang, C. “Targeting ultimate accuracy: Face recognition via deep embedding.” arXiv 2015, arXiv:1506.07310v4.
25. Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, Aleksander Madry. “How Does Batch Normalization Help Optimization?” arXiv:1805.11604v5 [stat.ML]
26. Parkhi O.M, Vedaldi A, Zisserman A. “Deep Face Recognition.” In Proceedings of the 2015 British Machine Vision Conference, Swansea, UK, 7–10 September 2015; pp. 41.1–41.12.
27. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. “Deep Residual Learning for Image Recognition.” arXiv:1512.03385 Dec 2015

Appendix

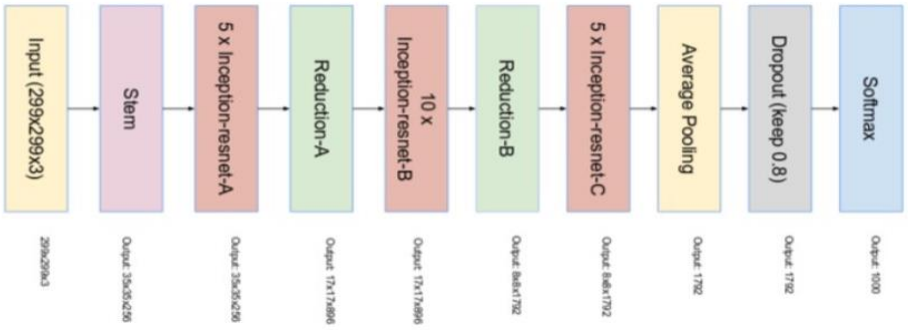


Figure 1 - Inception-ResNet v1 architecture

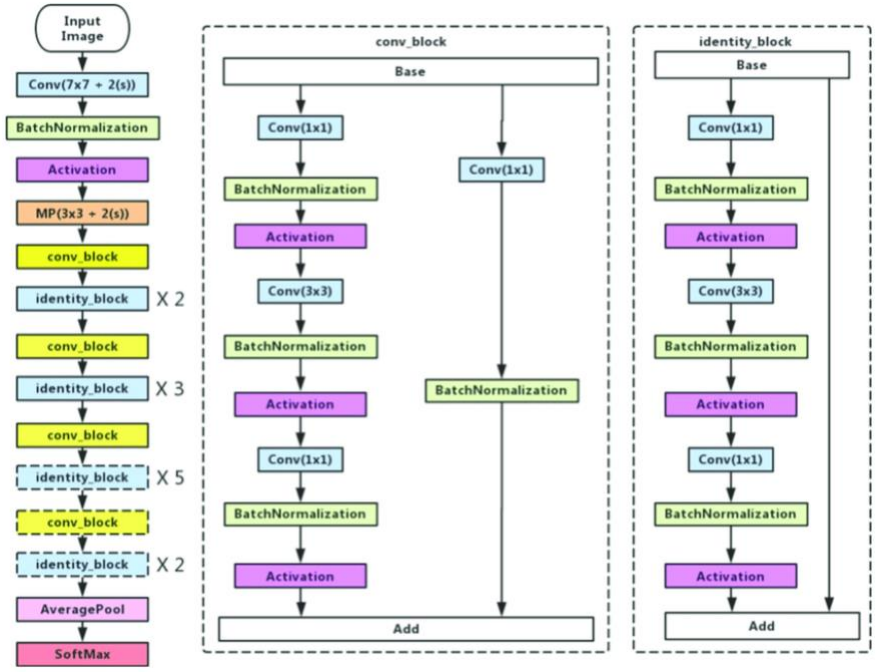


Figure 2: ResNet50 Architecture

Training Loss for Epoch:75, Margin=2; Learning Rate=0.01: