Milestone: Privacy-preserving Face Mask Detection using Few-shot Learning

Zeyan Liu 3001190

Project Team 10

[*zyliu@ku.edu*](mailto:zyliu@ku.edu)

***Abstract*—Face mask detection using Artificial Intelligence during the COVID period is a heated topic. The training of deep neural networks (DNNs) will require new datasets specially containing face masks, which raises privacy concerns of portrait usages. However, we want to address the privacy problems by showing that the task of faces mask detection can be generalized across different people conveniently using few-shot learning.**

1. INTRODUCTION

In the past decade, deep learning has gained incredible successes in a wide range of applications, fueling advances in all fields related to big data analysis. A representative of deep learning application is face identification. As neural networks become larger and deeper, model performance generally improves, but at the same time, model training requires significant amounts of data and expensive computational costs. During the COVID period, the face mask wearing becomes an important problem related to public health and disease control. However, large amounts of data is not always available for model training, especially considering human faces which will raise widespread concerns like privacy and copyright. Therefore, in this project, we want to solve the question: is it possible to train face mask detection model without sufficient data, e.g. using photos of one person or two?

Recently, many techniques have appeared towards the problem of data and computational sources in deep learning training, e.g. transfer learning and few-shot learning. Hopefully, we can address this problem using few-shot learning and transfer learning.

1. BACKGROUND
2. *Transfer Learning*

The idea of transfer learning is to borrow knowledge from source domains with sufficient data into target domains with highly limited data. Transfer learning can be further classified into multi-task learning, self-taught learning, domain adaptation and so on. In this project, we concentrate on the simple framework as fine-tuning pre-trained models. In this framework, Student models copy the network architectural structures and specific parameters of Teacher model obtained from previous large-scale training. Specifically, the Student model is initialized by copyingall layers of the Teacher model but the last layer. A final new fully-connected layer is added and trained for student classification task whose size matches the label space dimensionality of target domain. Among these layers, parameters of some first layers in Teacher model are completely reserved and directly reapplied in Student model. Thus these layers are called “frozen” layers and not for subsequent finetuning. For the other layers, the parameters are finetuned on the new data in the target domain for better performance.

Transfer learning successfully mitigate the general conflicts between shortage of training data and computing

devices for individuals and high resource requirements by well-performing neural networks with complicated structures, which provides convenience to small entities for quickly building up accurate deep learning models aiming at their own task.

1. *Siamese and Triplet Networks*

Siamese networks are two networks with shared structures and weights, and their weights are updated simultaneously. The goal of Siamese networks is to learn feature representations of two different input vectors so that similar images have higher proximity in the feature space while different images deviates far away. Technically, this is fulfilled by a comparative loss function that minimizes the distance of embeddings of inputs in the same category while maximizing the ones in different categories [12]. This is a common technique applied in few-shot learning with limited labelled data [13], or when inputs are highly identical in large part of input space [14]. A Triplet network follows exactly the similar discipline, but with three sub-networks. It performs the training on three images which are grouped into two input groups, including an anchor image, a positive image whose identity is the same as the anchor, and a negative image with a different identity. The overall objective of the network is to learn a feature space in which the distance between the positive and anchor images’ embeddings is smaller than the distance between the anchor and negative images’ ones.

1. APPROACH OVERVIEW

We will follow a pipeline of face detection algorithm, which typically consists of two phases: face detection and classification.

1. *Face Detection*

The first step is to detect and subtract the pixels of human faces from its background. Graphically, it is an operation of drawing bounding boxes around human faces. Clipped human faces provides the basis for further analysis. The detection can be two-stage including face recognition and alignment, or in one-stage manner. Common methods including HOG-based detectors, MTCNN [18] and YOLO [19].

1. *Mask Identification*

After clipping the human faces, the problem is simplified to classification on faces. In this project, we want to apply a binary classification: with or without masks. Technically, we wa nt to train a neural network to distinguish the human faces wearing masks from ones without masks. We can try to apply transfer learning to substitute the last layer with a N-to-2 fully-

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

connected layer and do some fine-tuning. Hopefully, this will saves us much time and computational resources for training.

1. *Few-shot Learning*

In Siamese Network, two parallel models sharing the same weights take pairs of images as input. One branch takes images of human faces with face masks as the baseline. The other branch takes the images without face masks. A contrastive loss is trained to maximize the distance between these two branches. In this way, the neural network is trained to specifically distinguish mask wearing.

Different from traditional neural network training, we can build a working model without much data, i.e. the images of human faces because the neural network doesn’t need to learn the complex features as in multi-label classification, but to approximate an optimal boundary of hyper-plane. In this way, we can train a model only on images of users who authorizes us, thus avoid privacy-sensitive portrait usage.

1. DATASET & EXPERIEMNT

For privacy purpose, the training dataset in this project will be the photos of a specific person, including ones with without masks. We will use several datasets for validation. Besides, I may handpicked pictures of human covering mouth with hands for cross-validation to address Malicious Attackers in V.

1. *Datasets*
   1. **Github Source.** [7] provides a dataset of labelled images of 5000 masked faces for 525 people and 90000 normal faces.
   2. **Simulated LFW Dataset**. A simulated dataset provided by [7] consists of over 20000 labelled images. In this dataset, the masks are manually manipulated and patched on the original images in LFW dataset.
   3. **Kaggle Face Mask Detection.** This dataset [8] contains 853 images of people in real-world belonging to the

3 classes: with masks, wihout masks and masks worn improperly. They also provide the clipped bounding boxes in the PASCAL VOC format.

1. *Additional Analysis*

Several technical analysis and discussions will be provided to this deep learning problem:

* **Saliency Analysis**. Some visualization tools, e.g. GradCAM, will be used to show the salient features learned by the deep neural networks.
* **Ablation Study**. The setting of parameters will be discussed, and the effects of parameter selection will be compared.
* **Pruning and Model Structure**. Ba sica lly, the project will start by transfer learning [17] from ResNet family

[16] as a baseline. However, based on the tasks in III and challenges in V, we may adjust the model structures to be lightweight or deeper.

1. *Preliminary Experiment Results*

Experiments are done on Simulated LFW dataset so far. Actually, an ideally privacy-preserving using only one single image is not practical due to the overfitting and the mechanism of mini-batch updating. Actually, a key observation is that the batch size effect convergence significantly. Batches of too small sizes will result in a great fluctuation in the direction of gradient descent and significantly slow the training process and converge. This problem cannot be nicely solved even with data augmentation.

However, data augmentation is still a useful tool to learn invariant features, which is the covering of the lower part of the face in this project. We will show in later part an ablation study of the data augmentation and hyperparameters.

The experiments are implemented on Ubuntu16.04 with Intel i9-9900k and Nvidia Geforce RTX2080Ti. The host model is InceptionResNetV1, which is the key component of FaceNet. The model parameters are pretrained on VGG Face Dataset. For convenience of training, I added some additional fully-connected-layers concatenated at the end to turn this problem into a binary problem. We have advantage of adding additional layers instead of finetuning the last layer for a mapping into 2-dimensional (for two labels). Firstly, we can nicely preserve the knowledge from teachers. Secondly, we no longer need to tweak the hyper-parameters of loss defined on representation vectors of the Siamese Networks. Instead, we turn it into a well-defined classification which can be well defined using conventional loss functions like cross-entropy.

The training set contains 16 pairs of examples. 8 of them are normal faces and 8 others are masked faces. During training, each pair are fed into the network with the label as whether they are from the same group or not. During validation period, over 20000 images are used. Experiment results show that we can have 85% of accuracy on validation set when trained only on only 16 pairs of images for only 10 epoches. This shows that our method is highly effectively.

We will show ablation study, saliency analysis, experiments on other datasets and pruning in formal final report.

1. CHANLLENGE

Considering the task we have clarified in III, we may expect a good performance immediately, as the fundamental problem is basically to build a neural network giving saliency to areas around mouth. Also, the feature spaces of mouth and masks are not difficult to distinguish because their sizes, shapes and colors are different. However, foreseen challenges are to solve if we want to apply our model in real-world scenarios:

* **Disturbing Factors.** The angle of human face and light conditions of the environment may significantly influence the model performance. People may also wear masks in an improper manner. A possible solution is to include images of different circumstances in the training dataset.
* **Robustness and Natural Adversarial Examples.** The colors and shapes of the masks may be very diverse in real life. In extreme case, there are even face masks for fun including patterns of human faces.
* **Malicious Attackers.** The model should prevent attackers from bypassing detection by covering their faces with common items like hands, books or smartphones when facing supervising camera. Intuitively, a triplet network is better for distinguishing three groups of targets: faces with masks, face without covering and face covered by fraudulent objects. However, this will increa se the difficulty of adversarial training, especially when the data amount is limited.
* **Machine Learning Fairness**. The model should be questioned if we are to apply few-shot learning to preserve privacy, as the model will be highly likely to have better performance on cases of similar gender, age and race with the training data.

REFERENCES

1. Meenpal, T., Balakrishnan, A., & Verma, A. (2019). Facial Mask Detection using Semantic Segmentation. 2019 4th International Conference on Computing, Communications and Security (ICCCS).
2. Chowdary G J, Punn N S, Sonbhadra S K, et al. Face mask detection using transfer learning of inceptionv3[C]//International Conference on Big Data Analytics. Springer, Cham, 2020: 81-90.
3. Chavda, A., Dsouza, J., Badgujar, S., & Damani, A. (2021). Multi- Stage CNN Architecture for Face Mask Detection. 2021 6th International Conference for Convergence in Technology (I2CT).
4. Suresh, K., Palangappa, M., & Bhuvan, S. (2021). Face Mask Detection by using Optimistic Convolutional Neural Network. 2021 6th International Conference on Inventive Computation Technologies (ICICT).
5. Singh, S., Ahuja, U., Kumar, M., Kumar, K., & Sachdeva, M. (2021). Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. Multimedia Tools and Applications, 80(13), 19753–19768.
6. Adrian Rosebrock. COVID-19: Face Mask Detector with OpenCV, K eras/TensorFlow, and Deep Learning. Pyimagesearch[. https://www.py](https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/) [imagesearch.com/2020/05/04/covid-19-face-mask-detector-with-open](https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/) [cv-keras-tensorflow-and-deep-learning/](https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/)https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset[.](https://github.com/prajnasb/observations/tree/master/experiements/data)
7. Face Mask Detection Dataset. Kaggle. [https://www.kaggle.com/andre](https://www.kaggle.com/andrewmvd/face-mask-detection) [wmvd/face-mask-detection](https://www.kaggle.com/andrewmvd/face-mask-detection).
8. Mask Wearing Dataset. Roboflow[. https://public.roboflow.com/object](https://public.roboflow.com/object-detection/mask-wearingTraining)

[-detection/mask-wearingTraining](https://public.roboflow.com/object-detection/mask-wearingTraining) f

1. Nawaz, Faisal; Khan, Wasiq; Yasen, Salwa; Hussain, Abir (2020), “F ace Mask Detection Video Dataset”, Mendeley Data, V1, doi: 10.1763 2/v3kry8gb59.1 <https://data.mendeley.com/datasets/v3kry8gb59/1>
2. Sethi S, Kathuria M, Kaushik T. Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread[J]. Journal of Biomedical Informatics, 2021, 120: 103848.
3. Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for oneshot image recognition[C]//ICML deep learning workshop. 2015, 2.
4. Bromley J, Guyon I, LeCun Y, et al. Signature verification using a” siamese” time delay neural network[J]. Advances in neural information processing systems, 1993, 6: 737-744.
5. Dey S, Dutta A, Toledo J I, et al. Signet: Convolutional siamese network for writer independent offline signature verification[J]. arXiv preprint arXiv:1707.02131, 2017.
6. Selvaraju R R, Cogswell M, Das A, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization[C]//Proceedings of the IEEE international conference on computer vision. 2017: 618-626.
7. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
8. Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2009, 22(10): 1345-1359.
9. Xiang J, Zhu G. Joint face detection and facialexpression recognition with MTCNN[C]//2017 4th international conference on information science and control engineering (ICISCE). IEEE, 2017: 424 -427.
10. Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.