INTRODUCTION TO ARTIFICIAL INTELLIGENCE

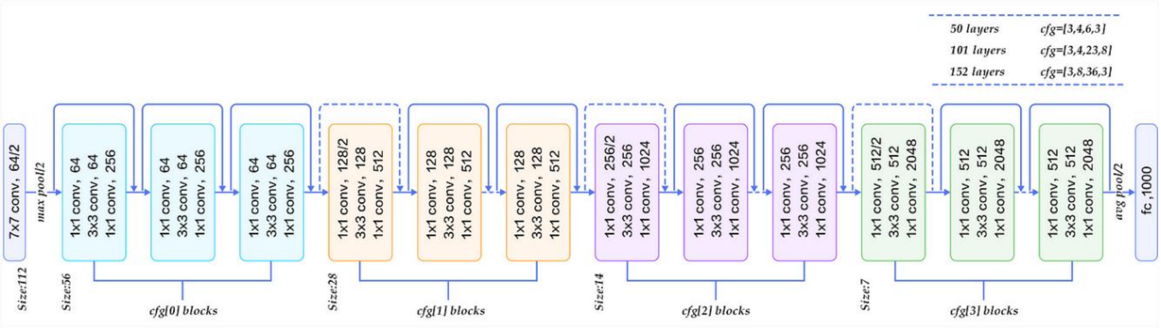
ICPBL project – Mask detection

2021038431 Sumin Lee

2022085241 Seokhyun Seo

1. ResNet-50 Architecture

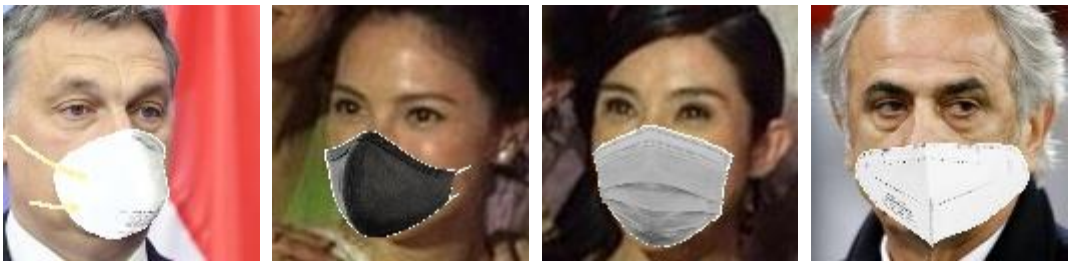
ResNet-50 (Residual Network-50) is a deep convolutional neural network widely used for image classification tasks. It is composed of 50 layers, including 49 convolutional layers and one fully connected output layer. The most distinctive feature of ResNet-50 is the introduction of residual connections, also known as skip connections. Instead of learning the direct mapping from input to output, each block learns a residual function F(x)=H(x)−x, allowing the network to effectively bypass unnecessary transformations and improve gradient flow during backpropagation. This design significantly alleviates the vanishing gradient problem and enables successful training of very deep networks.



The architecture begins with a 7×7 convolution followed by max pooling, and consists of four residual stages made up of 1×1, 3×3, and 1×1 convolution blocks with identity shortcuts. The number of filters increases in deeper layers, allowing the network to capture complex and abstract features. In our implementation, we used the torchvision.models.resnet50 module and replaced the final fully connected layer with a custom layer tailored for binary classification (mask vs. no mask). This architecture balances depth and efficiency, making it well-suited for our classification task involving facial mask detection.

2. Synthetic Data Generation

To compensate for the lack of masked facial images in the original dataset, we utilized MaskTheFace, an open-source tool that automatically applies synthetic masks to face images using facial landmark detection. This process generates realistic masked images by overlaying mask textures on detected facial regions. We applied MaskTheFace to both the training and validation sets, targeting only the "not_wearing_mask" images. To ensure consistency and improve training performance, we limited the types of synthetic masks to four commonly used designs: N95, KN95, surgical, and cloth masks.



By augmenting the dataset with these masked images, we achieved a more balanced distribution between masked and unmasked faces, which is critical for binary classification. This synthetic augmentation helped the model generalize better to real-world conditions where mask styles vary. Additionally, we visually verified the synthetic images and discarded those with misaligned masks or detection errors.

3. Data Preprocessing and Augmentation

To enhance model robustness and prevent overfitting, we applied a set of data preprocessing and augmentation techniques to both the training and validation datasets. All input images were resized and center-cropped to a fixed dimension of 112×112 pixels to ensure consistency across batches. We also normalized the pixel values using mean 0.5 and standard deviation 1.0, which stabilizes gradient updates during training. This normalization was applied uniformly across the dataset using PyTorch’s transforms.Normalize.

In addition to preprocessing, we introduced data augmentation techniques to increase variability in the training set. Specifically, we applied random rotations up to 30 degrees using transforms.RandomRotation(30). This helped the model learn rotational invariance and generalize better to different head poses.

4. Model Training

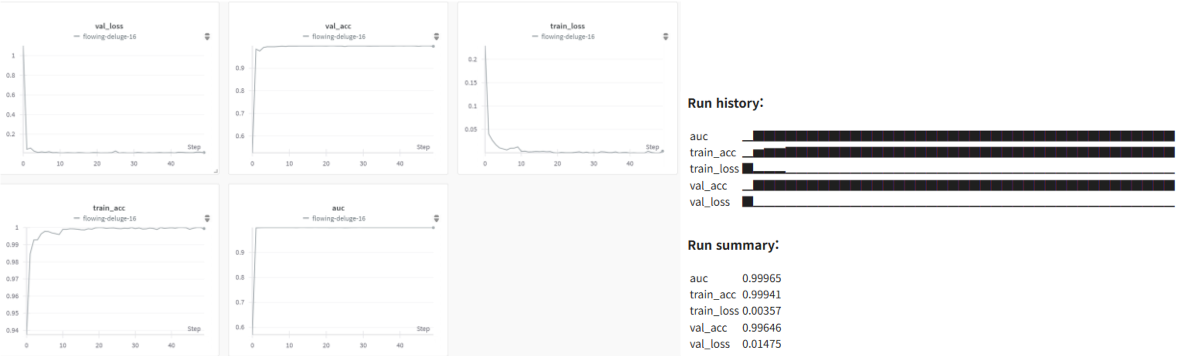
We trained the model using ResNet-50 as the backbone, with the final fully connected layer modified to output a single logit for binary classification (masked vs. unmasked). The model was implemented using PyTorch and trained on GPU via Google Colab. We used the Adam optimizer with a learning rate of 0.01, and the loss function was Binary Cross Entropy with Logits (BCEWithLogitsLoss), which is suitable for binary classification tasks involving raw logits. A learning rate scheduler (MultiStepLR) was applied with milestones at epochs 10 and 20, and a decay factor of 0.5 to fine-tune learning as training progressed.

The model was trained for 50 epochs with a batch size of 64. During each epoch, we calculated both training and validation accuracy and loss. To prevent exploding gradients, we applied gradient clipping with a max norm of 1.0. The model achieving the lowest validation loss during training was saved using torch.save(model.state_dict()), and this saved checkpoint was later used for evaluation. Our training loop also included real-time logging to Weights & Biases (WandB), allowing us to monitor performance metrics such as accuracy, loss, and AUC throughout the training process.

5. Performance Visualization using WandB

To track and analyze model performance in real time, we integrated Weights & Biases (WandB) into our training pipeline. Throughout training, WandB recorded key metrics such as training accuracy and loss, validation accuracy and loss, and Area Under the ROC Curve (AUC) for each epoch. These metrics were visualized through interactive line plots, which helped us identify the optimal epoch and monitor overfitting or underfitting trends. By observing the validation curves, we selected the best-performing checkpoint (model_epoch18.pth) where validation accuracy peaked at 99.65% and AUC reached 0.99965.

In addition to line graphs, we plotted the Receiver Operating Characteristic (ROC) curve to assess the model's discriminative ability. The ROC curve maintained a near-perfect shape, confirming that the model consistently distinguished between masked and unmasked faces. All visualizations were exported from WandB and included in this report to support our analysis. These graphs demonstrate the effectiveness of our training strategy and serve as clear evidence of model performance.

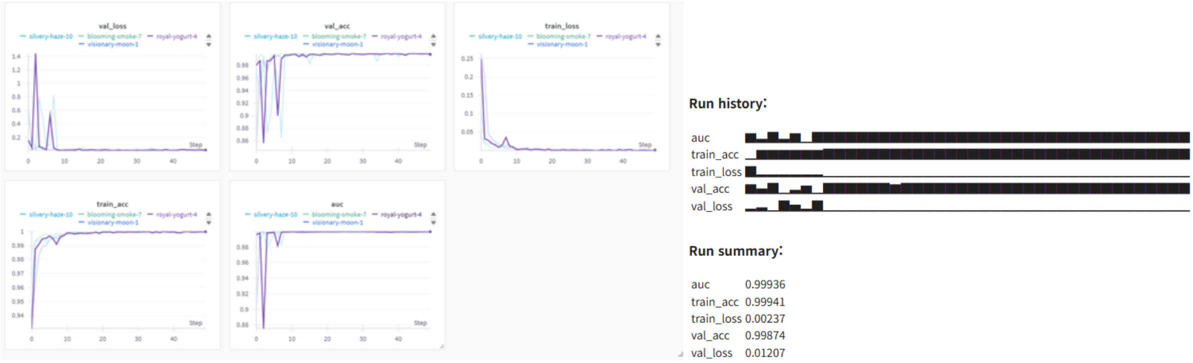


Run summary:

```
auc        0.99965
train_acc  0.99941
train_loss 0.00357
val_acc    0.99646
val_loss   0.01475
```

6. Ablation Study & Discussion

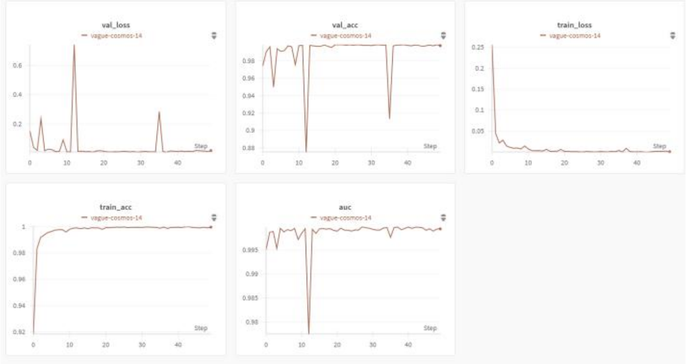
To better understand the impact of different architectural and training choices, we conducted two ablation studies. First, we compared ResNet-50 with and without pretrained weights. While both models achieved high performance, the pretrained version slightly outperformed the non-pretrained one, achieving a validation accuracy of 99.94% vs. 99.87% and AUC of 0.99965 vs. 0.99936. This suggests that pretrained weights can still offer marginal improvements, especially in convergence stability.

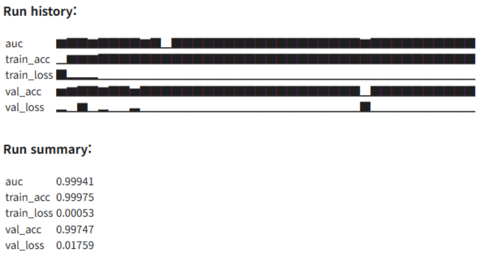


Run summary:

```
auc        0.99936
train_acc  0.99941
train_loss 0.00237
val_acc    0.99874
val_loss   0.01207
```

The second study involved replacing ResNet-50 with a deeper model, ResNet-152, while keeping all other training settings identical. Interestingly, the ResNet-152 model achieved slightly higher performance, with a validation accuracy of 99.75% and an AUC of 0.99941. However, the improvement was marginal, and the increased model size did not lead to substantial gains. This suggests that deeper architectures would be helpful for binary tasks when a well-performing baseline are constructed.



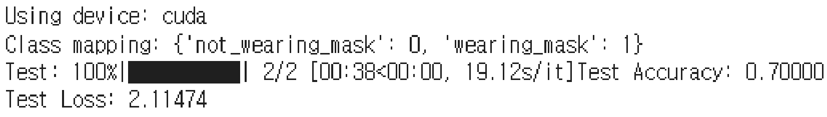


```
Run history:

auc
train_acc
train_loss
val_acc
val_loss

Run summary:

auc        0.99941
train_acc  0.99975
train_loss 0.00053
val_acc    0.99747
val_loss   0.01759
```

7. Qualitative Evaluation

To evaluate the model's real-world applicability, we conducted qualitative testing using images outside the training and validation datasets. We selected celebrity photos and personal photographs that were not included in the original dataset. These images were passed through the trained ResNet-50 model, and the predictions were highly accurate and intuitive. The model successfully detected masks of various styles and colors, including partial occlusions and non-standard angles, demonstrating strong generalization ability beyond the training data.

In addition, we observed that the model could accurately classify challenging cases, such as loosely worn masks or small face regions. Visual results confirmed that the model was not overfitted to the training data distribution and could handle diverse real-world conditions. These examples validate the practicality and robustness of our mask classification model in uncontrolled environments. The relatively high test loss is likely due to the small sample size and domain shift introduced by using celebrity photos, which may differ in lighting, pose, or resolution from the training dataset.



```
Using device: cuda
Class mapping: {'not_wearing_mask': 0, 'wearing_mask': 1}
Test: 100%|██████████| 2/2 [00:38<00:00, 19.12s/it]Test Accuracy: 0.70000
Test Loss: 2.11474
```

- **References**

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
https://doi.org/10.1109/CVPR.2016.90

[2] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019, pp. 8024–8035.
https://papers.nips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[3] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.
https://doi.org/10.1109/CVPR.2009.5206848

[4] A. Anwar, A. Raychowdhury, and M. Shah, "Masked Face Recognition for Secure Authentication," *arXiv preprint arXiv:2008.11104*, 2020.
https://arxiv.org/abs/2008.11104