



Multiclass AI-Generated Deepfake Face Detection

Zhakhayeva Zhanel, Ayan Igali,
Kaisagaliyev Miras, Torekhan Erdauit,
Salikhodzhaeva Sitora

Team members

1. Igali Ayan: **Team leading, researching, Vision Transformer model**
2. Zhakhayeva Zhanel: **Coding, ResNet101 and VGGNet16 models**
3. Torekhan Erdauit: **Coding, VGGNet19 model**
4. Salikhodzhaeva Sitora: **Dataset preparation, presentation**
5. Kaisagaliyev Miras: **Coding, ResNet50 model**

Why this topic?



AI models/bloggers



text-to-image



text-to-video

Showing 1-25 of 280 results for **AI generated face recognition** ×

Showing 1-25 of 125 results for **AI generated face classification** ×

Showing 1-25 of 3,733 results for **COVID Classification** ×

Showing 1-25 of 4,406 results for **covid detection** ×

Showing 1-25 of 1,826 results for **COVID Diagnosis** ×

screenshots from ieeexplore

Research on what we based



Article

Multiclass AI-Generated Deepfake Face Detection Using Patch-Wise Deep Learning Model

Muhammad Asad Arshed ^{1,*}, Shahzad Mumtaz ², Muhammad Ibrahim ², Christine Dewi ^{3,*},
Muhammad Tanveer ¹ and Saeed Ahmed ^{1,4}

Used Neural network architectures:

1. VGG16
2. ResNet50
3. ViT

Their objective was to show that transformer based models (Vision Transformers) are better than traditional models.

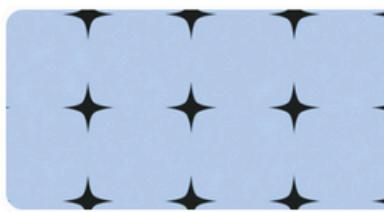
Model	Train Accuracy	Validation Accuracy	Accuracy	Precision	Recall	F1
ResNet-50	0.77	0.78	0.77	0.80	0.77	0.78
VGG-16	0.95	0.93	0.94	0.94	0.94	0.94
Proposed	0.99	0.99	0.99	0.99	0.99	0.99

Results of authors

Datasets

140k Real and Fake Faces

70k real faces (from Flickr) and 70k fake faces (GAN-generated)



140 thousand images (70k StyleGAN, 70 real)

Synthetic Faces High Quality (SFHQ) part 4

125K curated 1024x1024 face images. StyleGAN2 encoding of Stable Diffusion 2.1



125 thousand of AI-gen faces with StyleGan2

We took 10k from each class (30k overall) and upload it to Google Drive for convience

```
normalize = Normalize(mean=mean, std=std)
_train_transforms = Compose(
    [
        RandomResizedCrop(224),
        RandomHorizontalFlip(),
        ToTensor(),
        normalize,
    ]
)
```

```
_val_transforms = Compose(
    [
        Resize(224),
        CenterCrop(224),
        ToTensor(),
        normalize,
    ]
)
```

How we augmented data

Problem that we faced

1. Not enough computational power

that's all.

So we need to decrease dataset to 2500 image for each class (7500 overall)

Серверный ускоритель Python 3 на базе Google Compute Engine ((GPU)).
Показано потребление ресурсов в период с 12:07 до 12:09

Оперативная память системы
11.5 / 12.7 GB



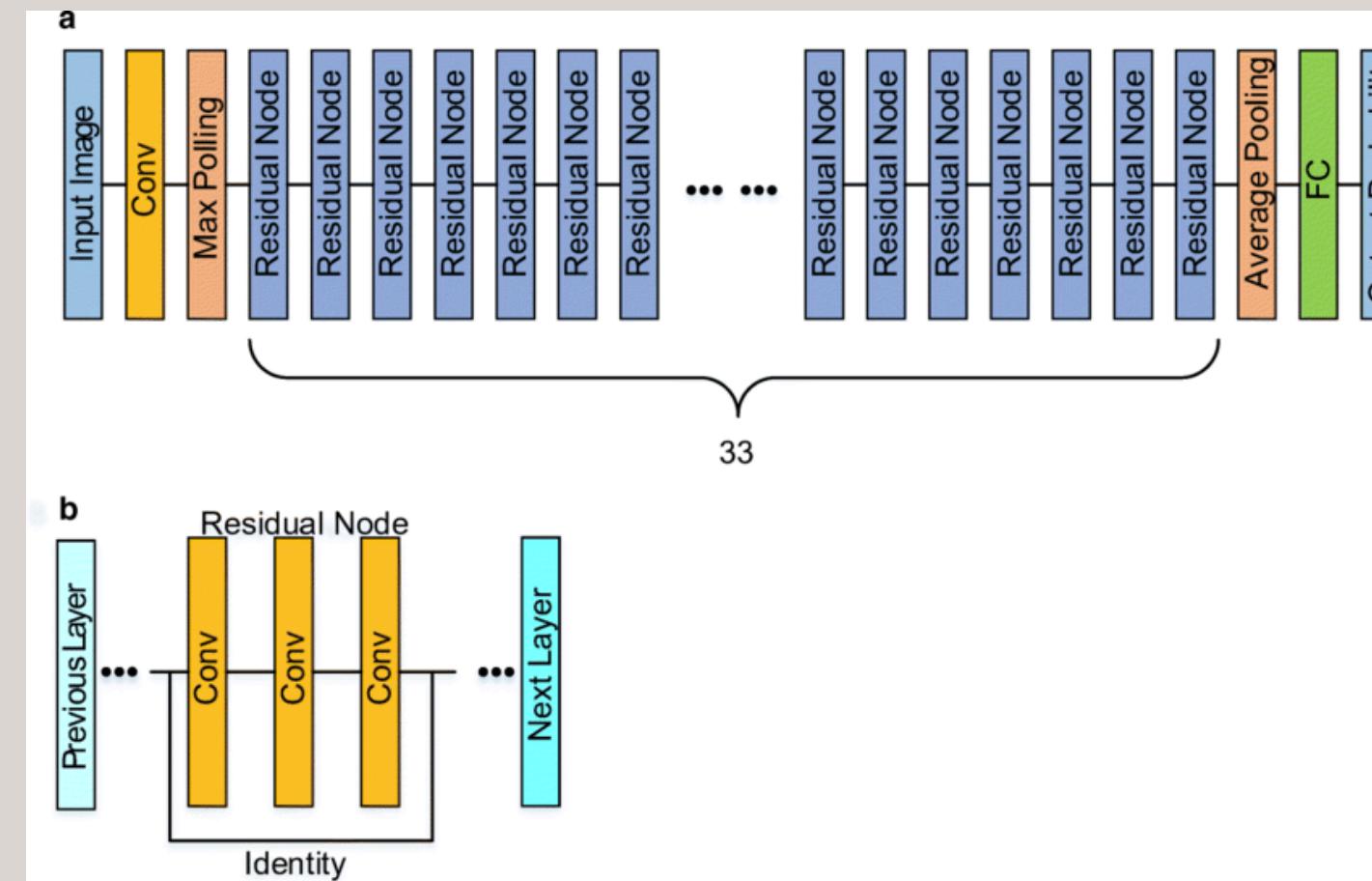
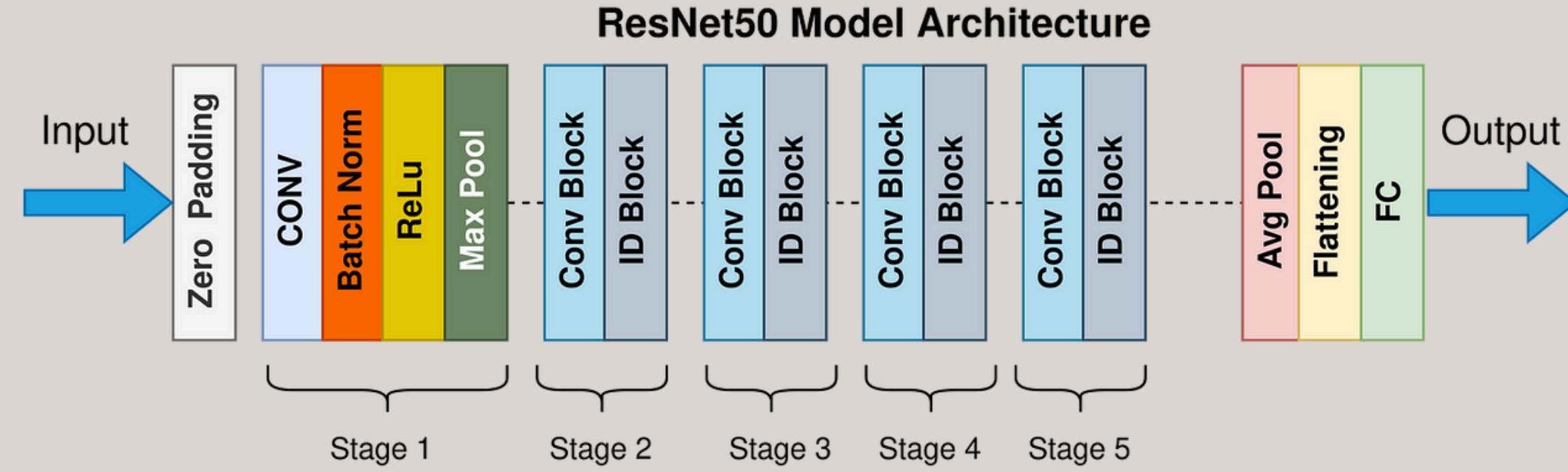
Model that we used

- 1. Vision Transformers**
- 2. ResNet50**
- 3. ResNet101**
- 4. VGGNet16**
- 5. VGGNet19**

Parameters

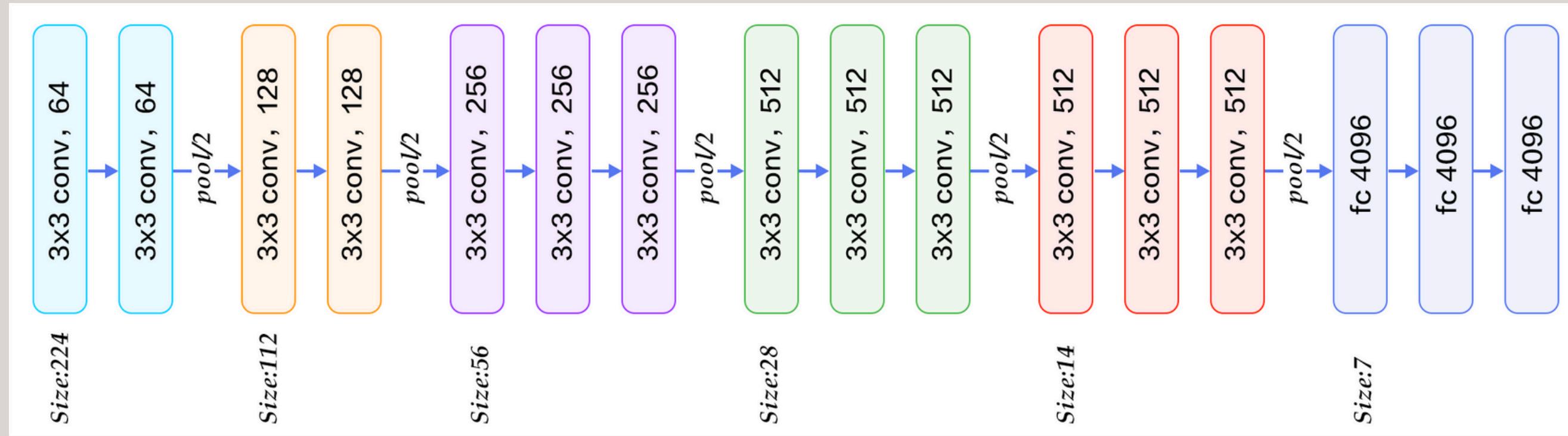
- 1. Epochs = 10**
- 2. Batch size = 16**
- 3. Learning rate = 0.0002**
- 4. Optimizer = AdamW**
- 5. Train size = 70%**
- 6. Validation size = 10%**
- 7. Test size = 20%**

ResNet

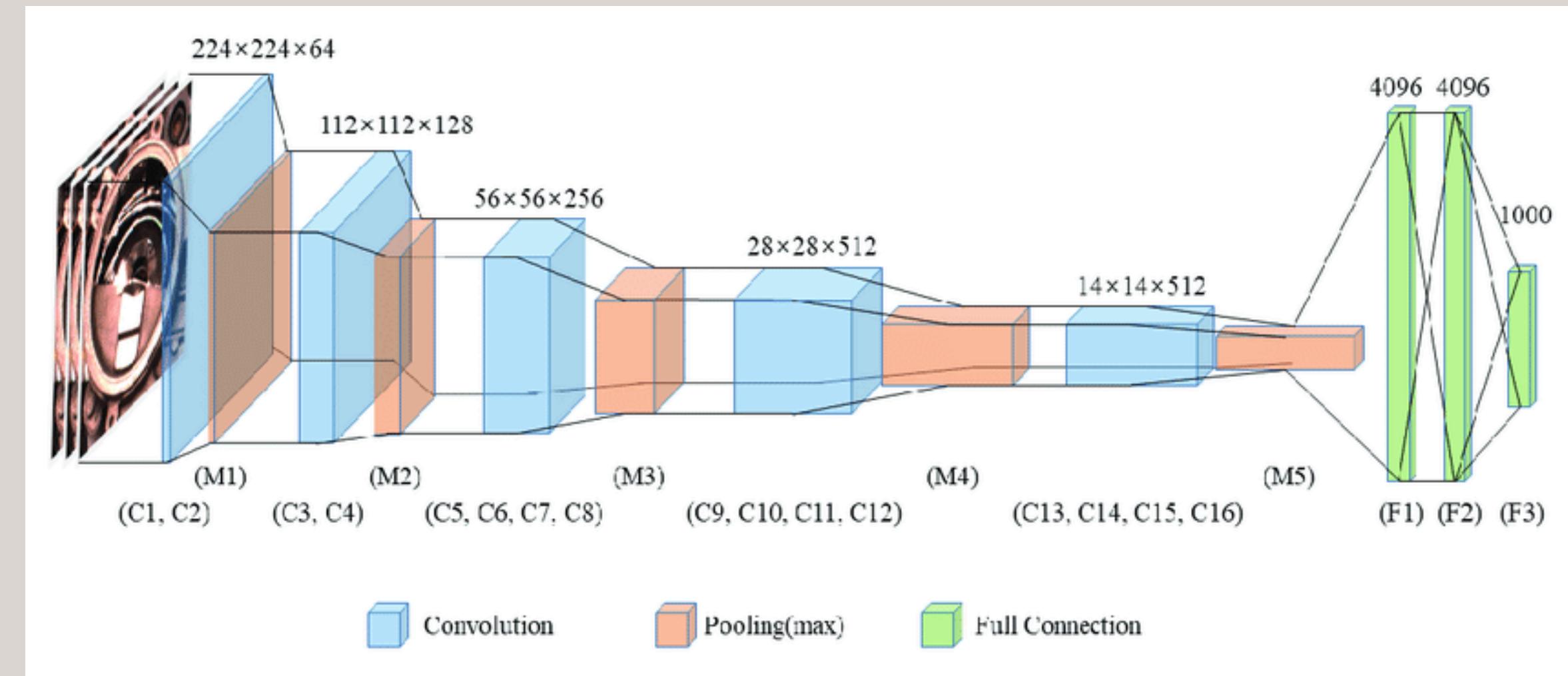


ResNet101

VGGNet

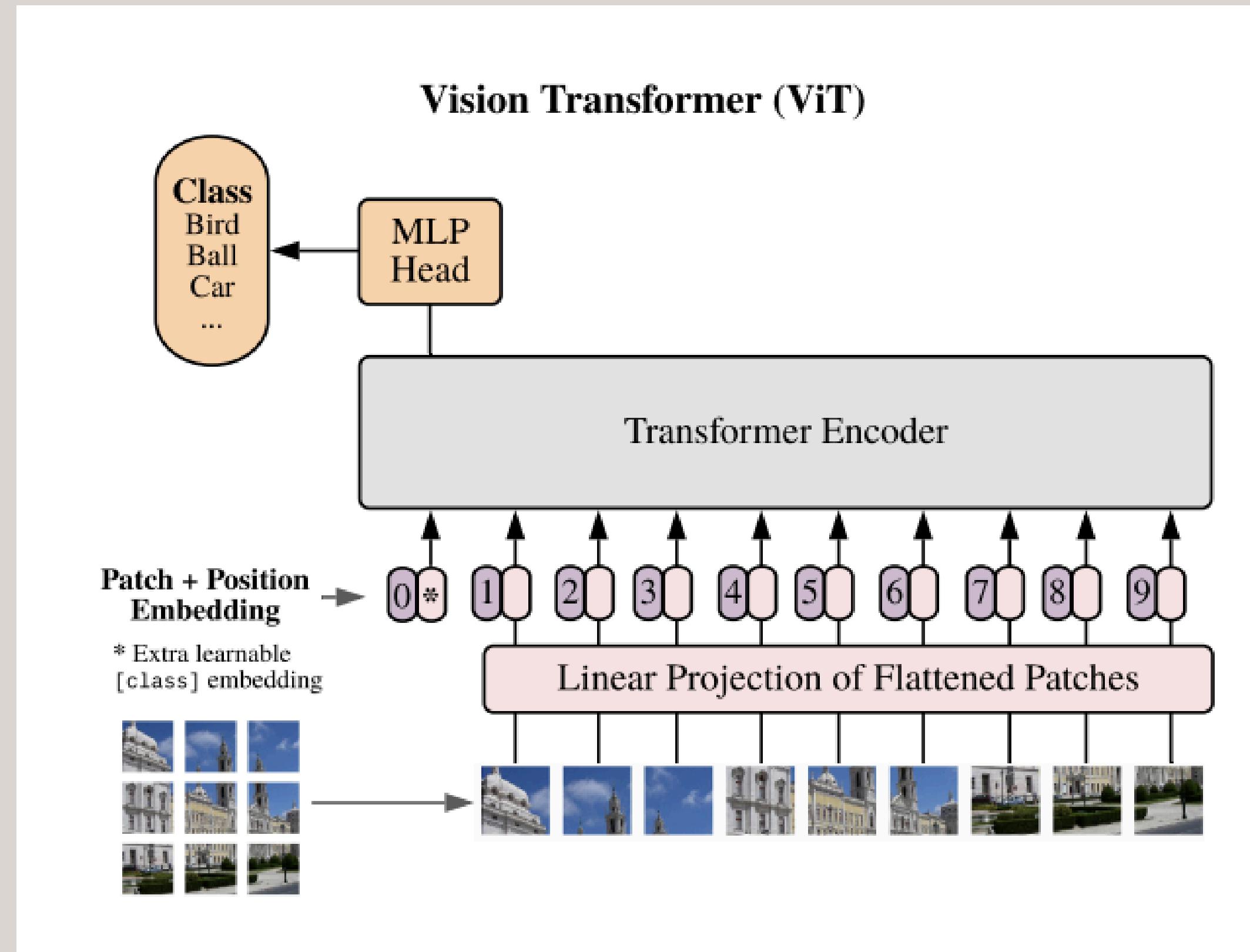


VGGNet16



VGGNet19

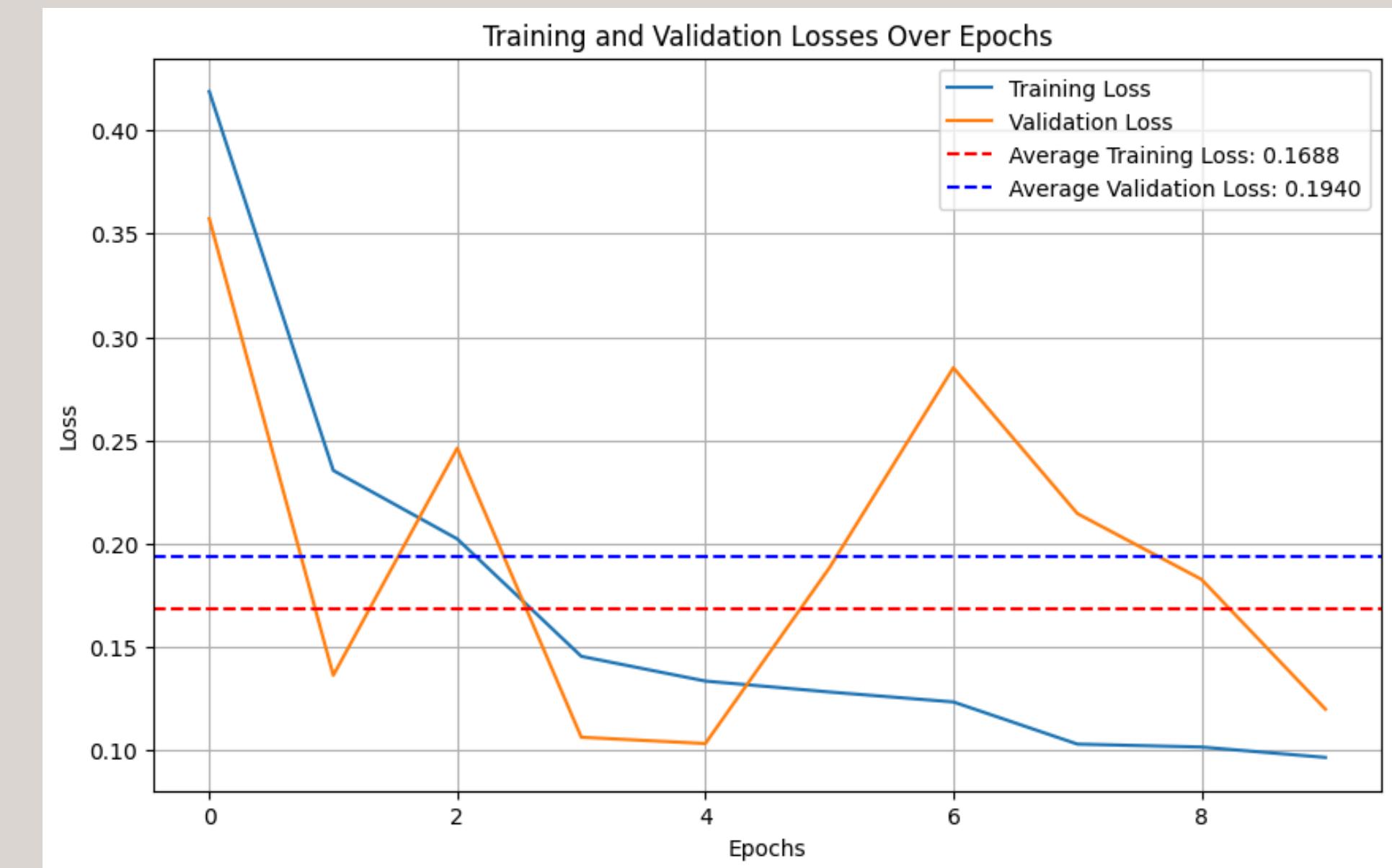
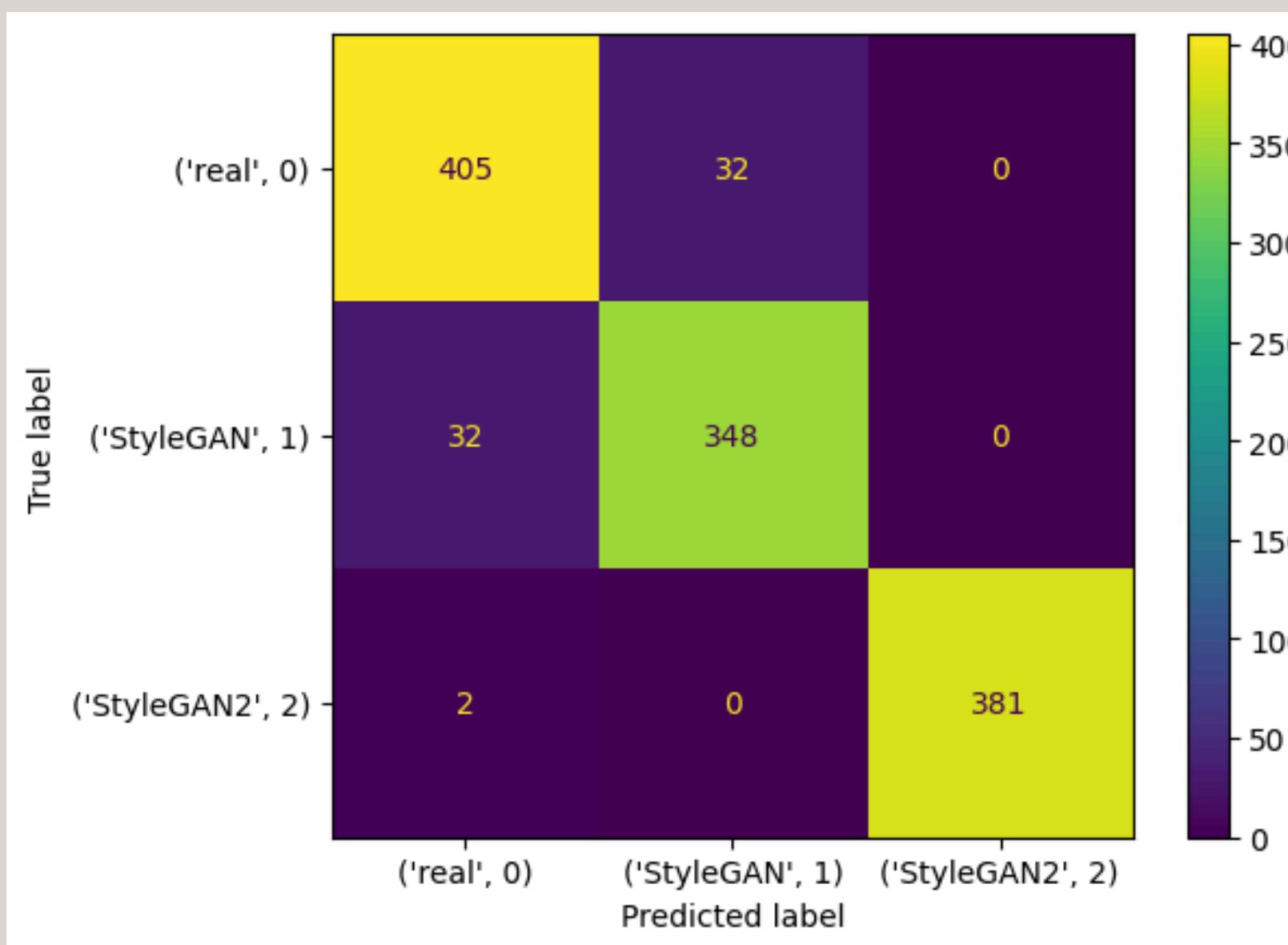
Vision Transformers

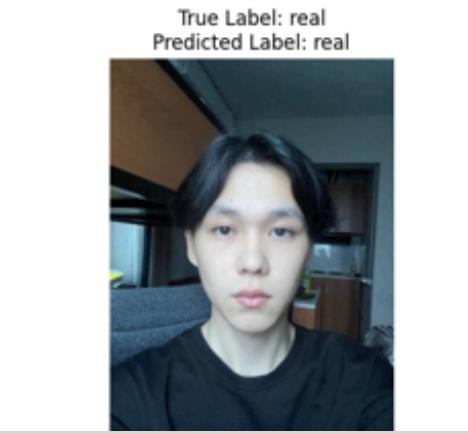
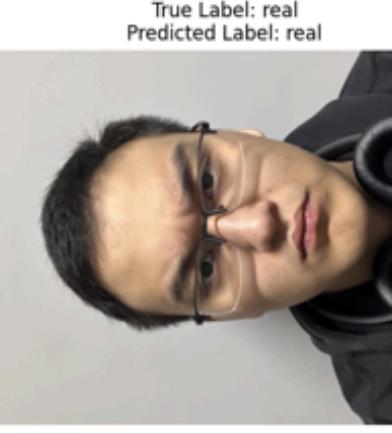
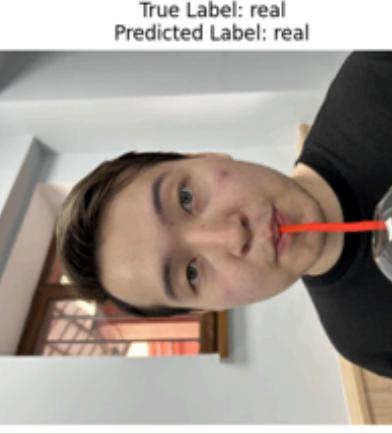
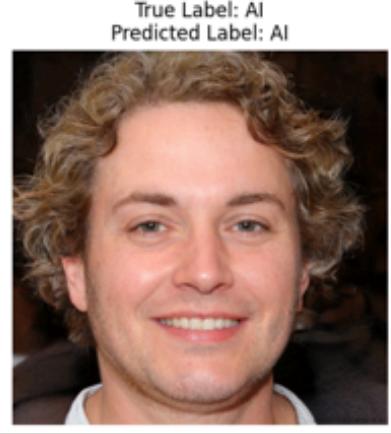
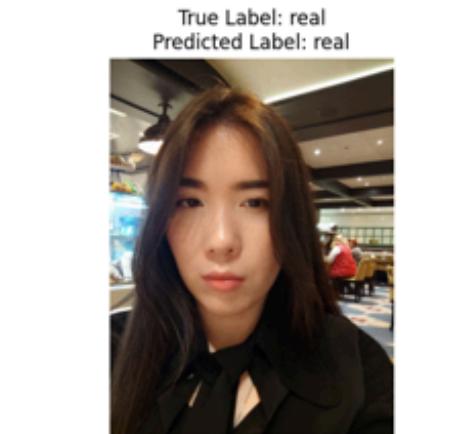
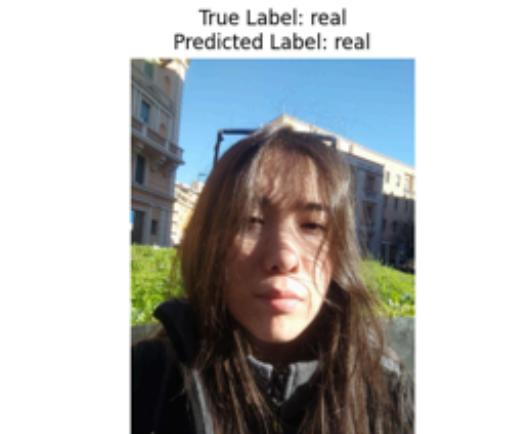
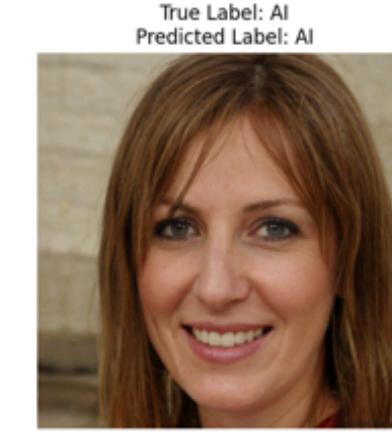
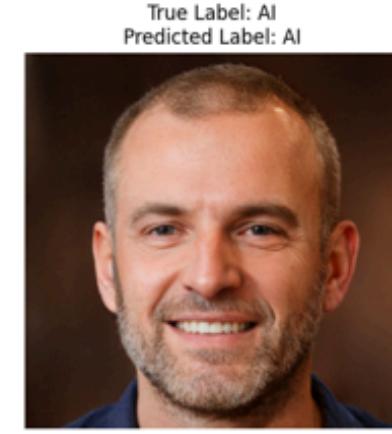
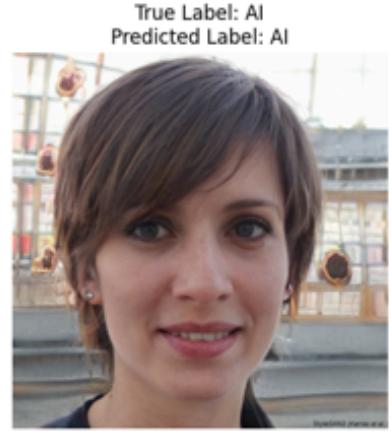
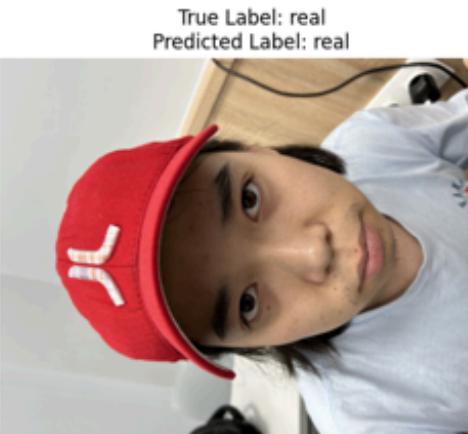
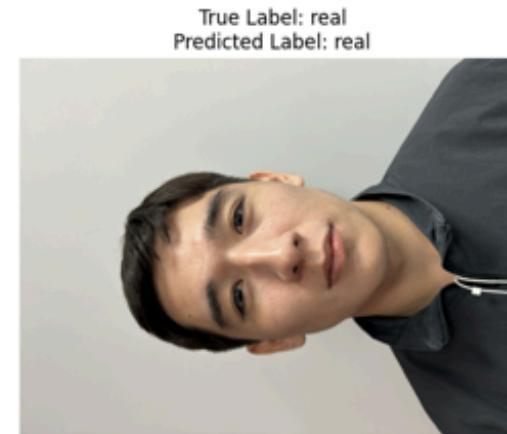
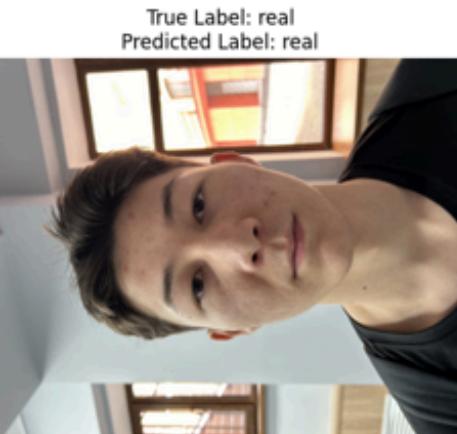
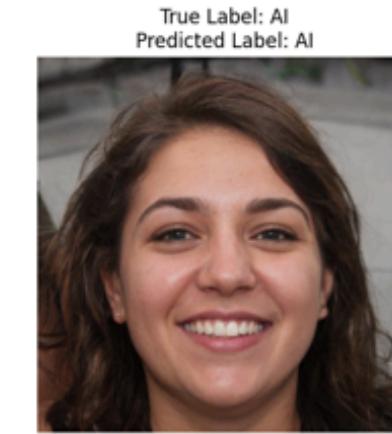
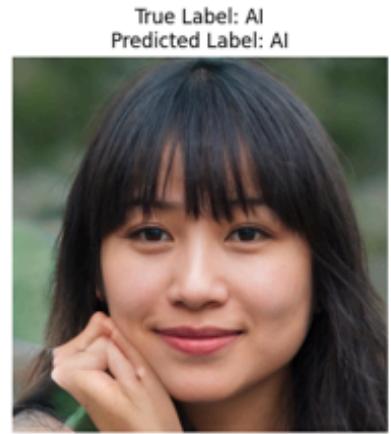
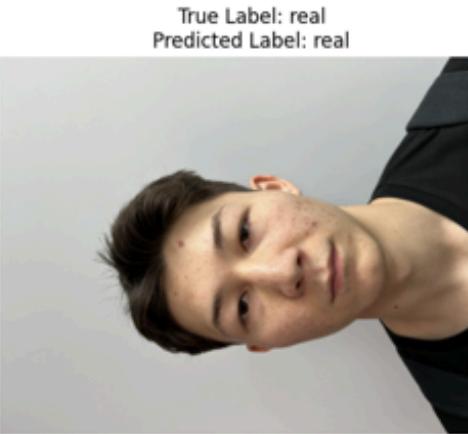
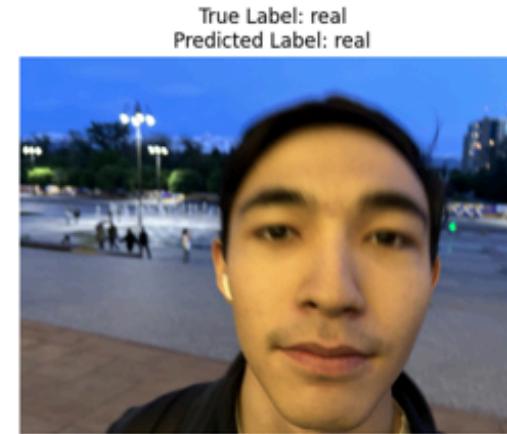
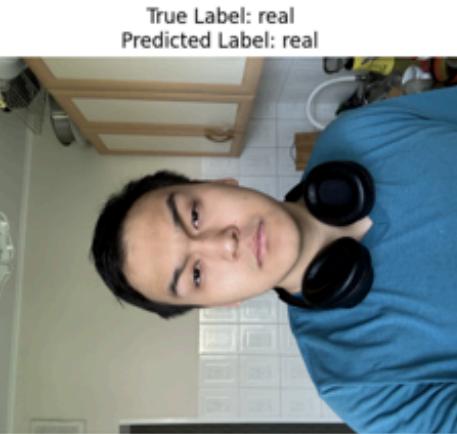
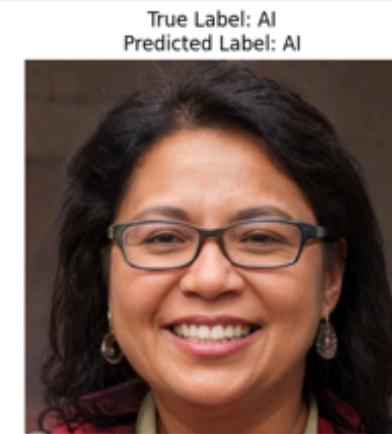
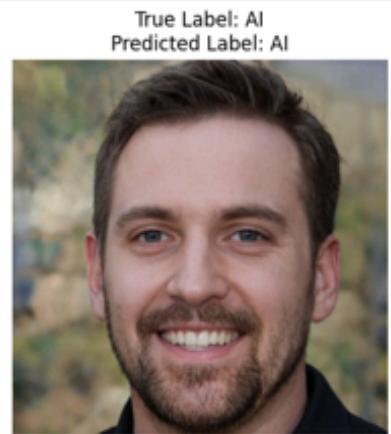


Results

ResNet50

Accuracy	F1-score	Precision	Recall	Val acc
95%	0.9459	0.9461	0.9458	95%

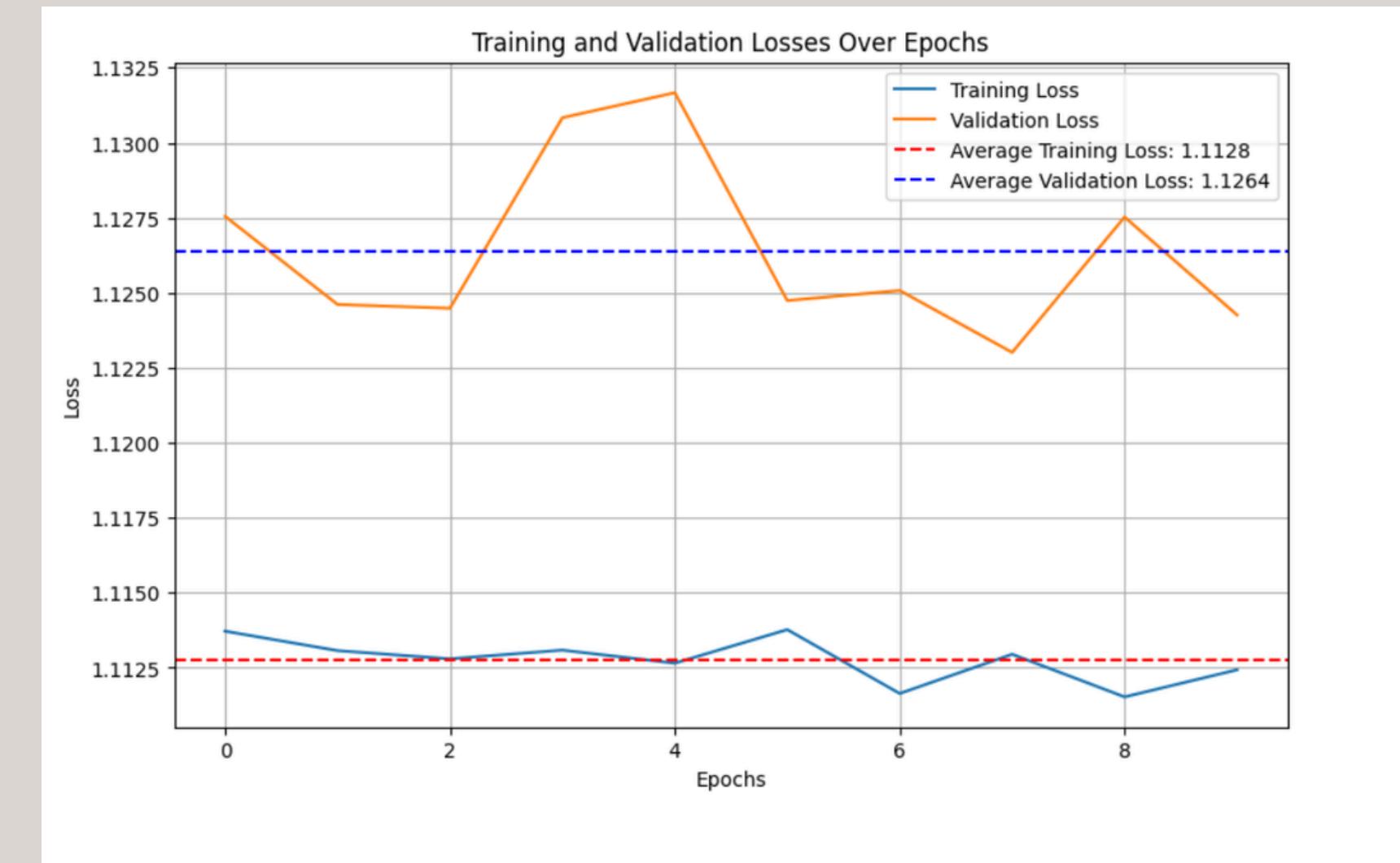
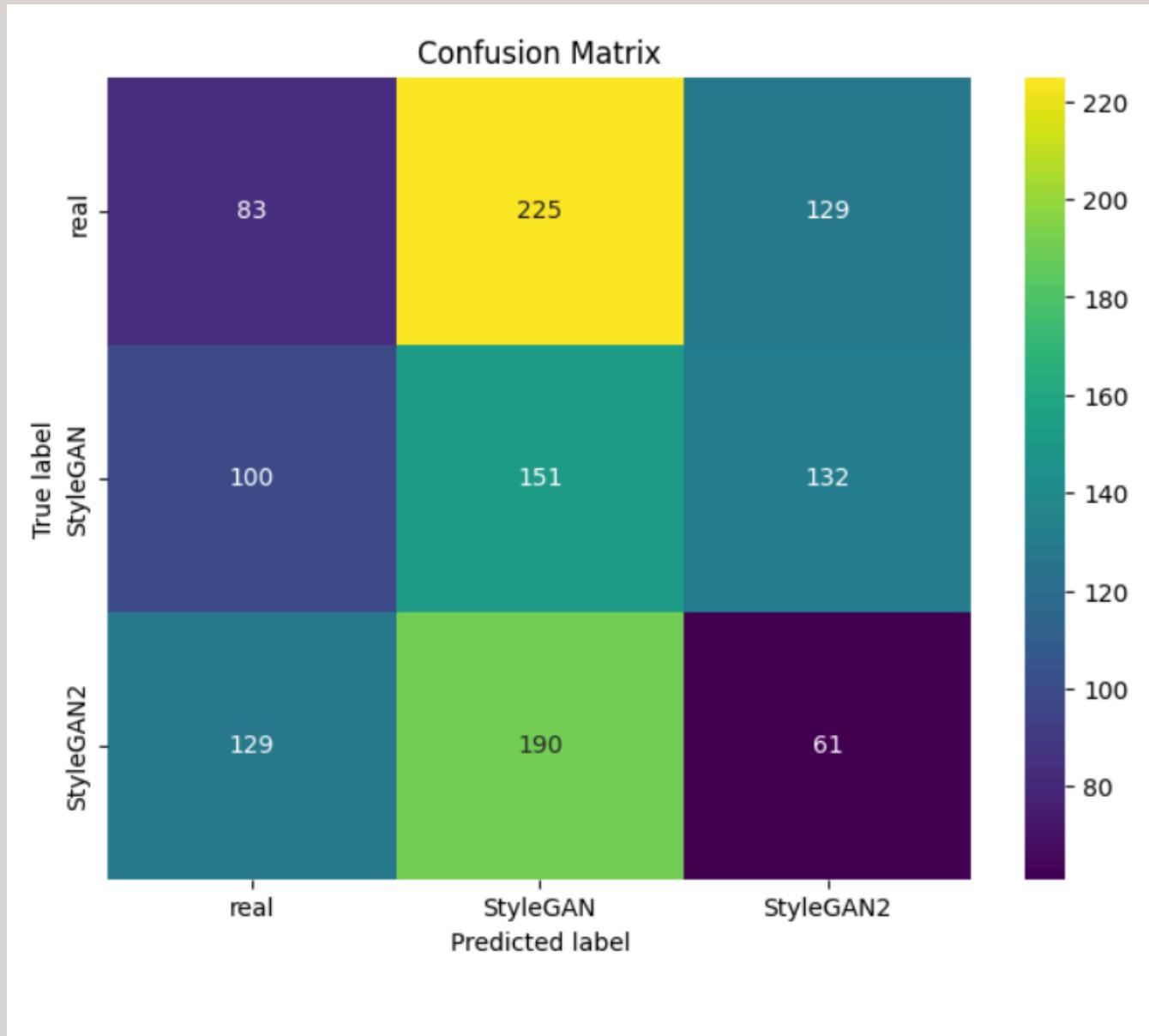


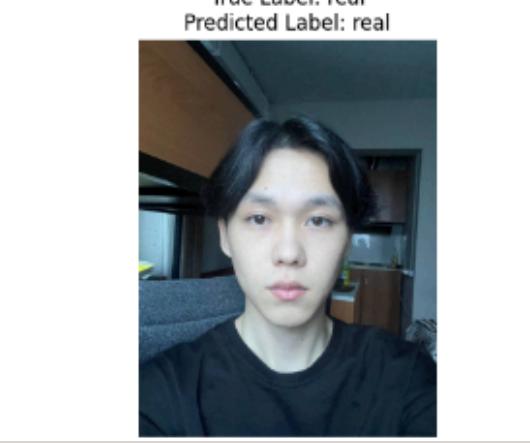
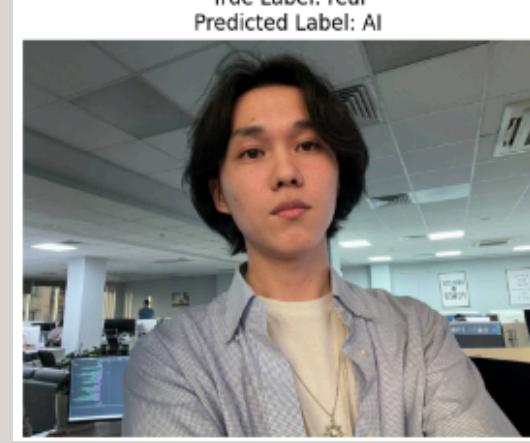
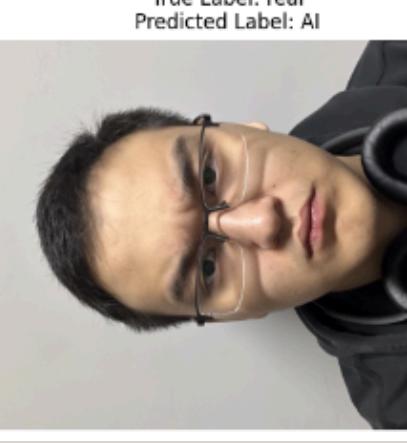
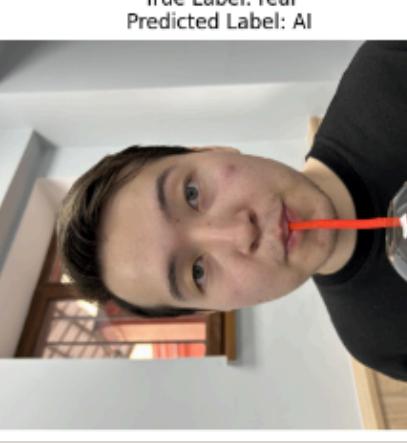
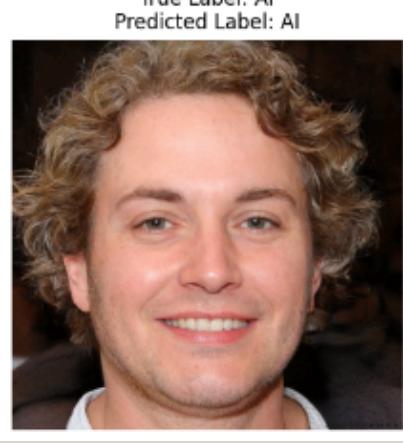
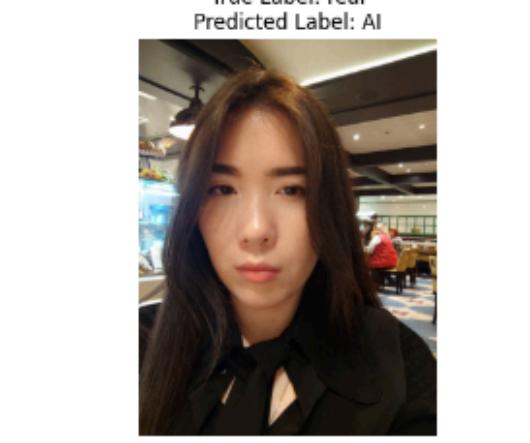
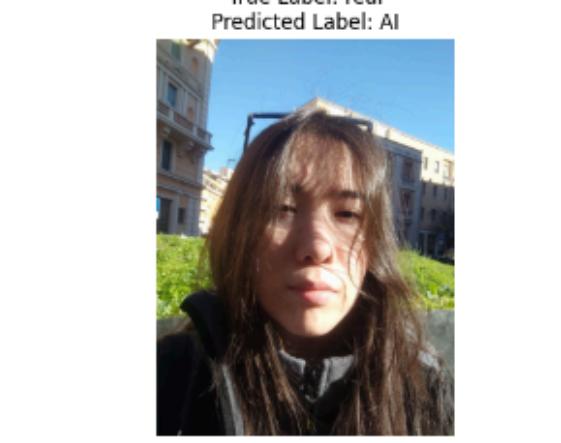
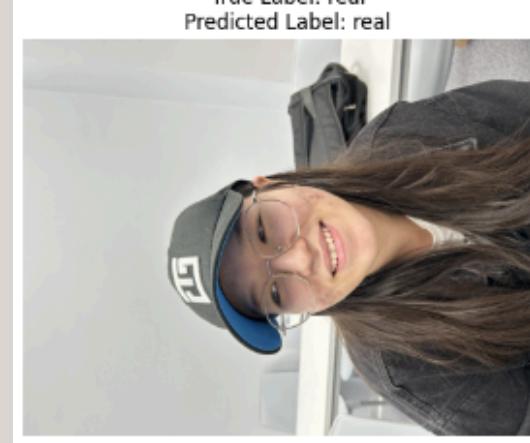
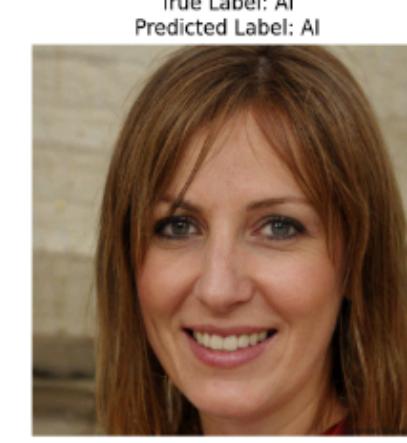
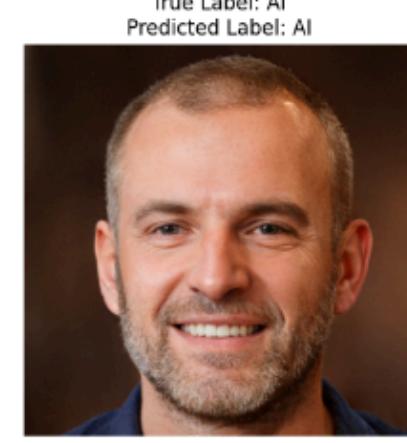
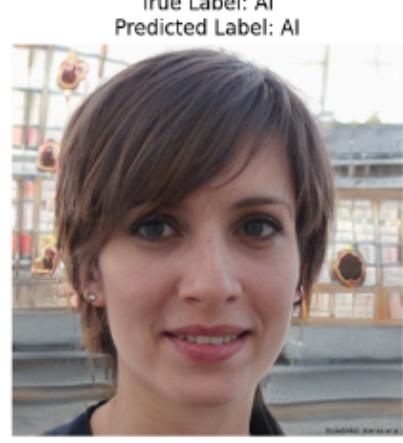
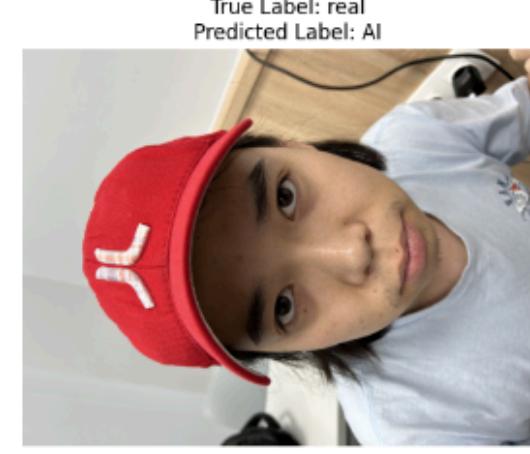
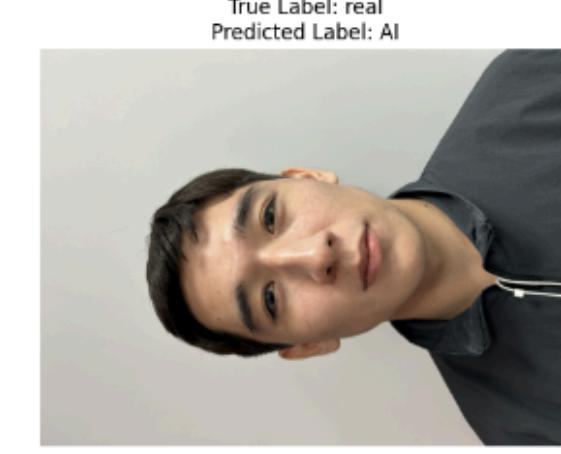
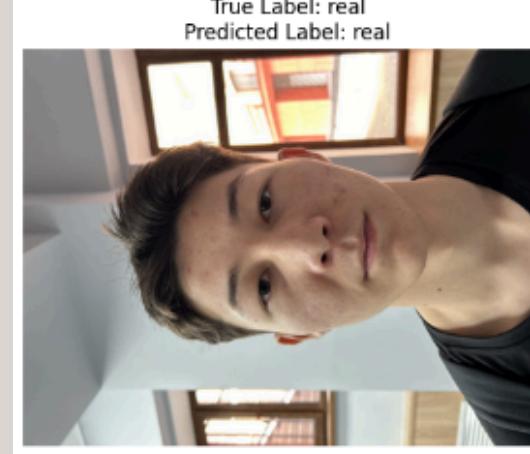
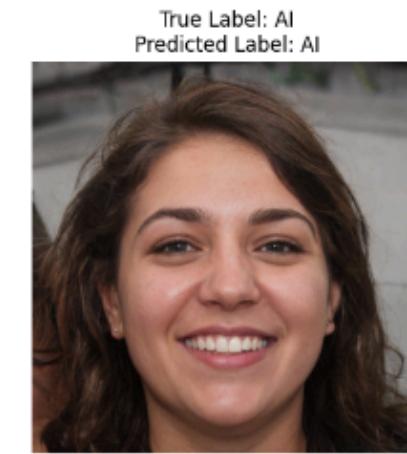
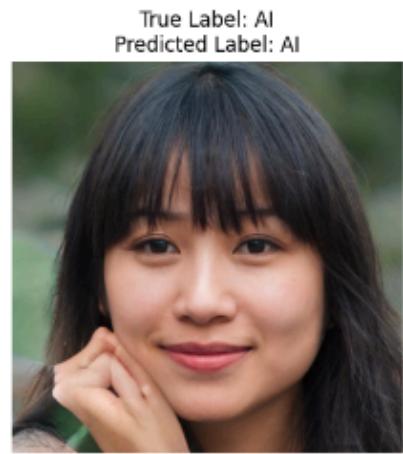
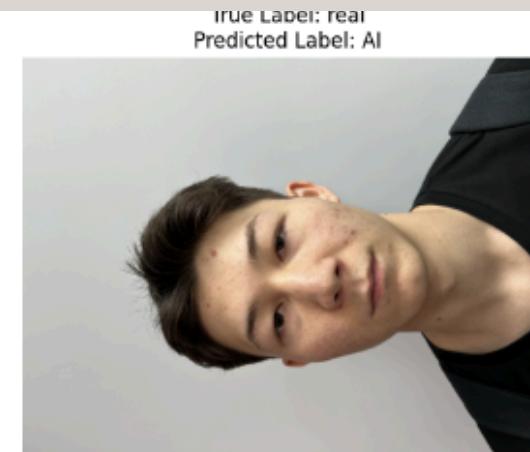
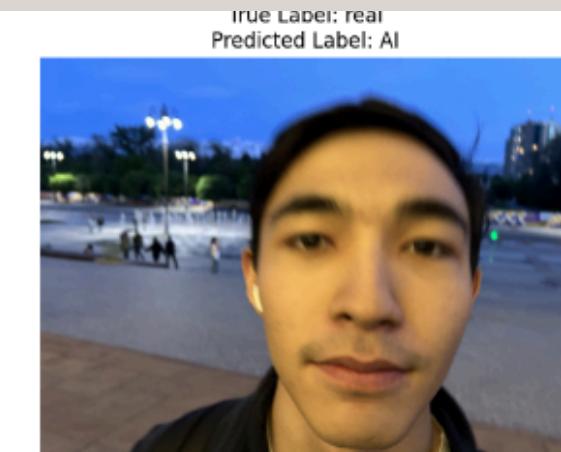
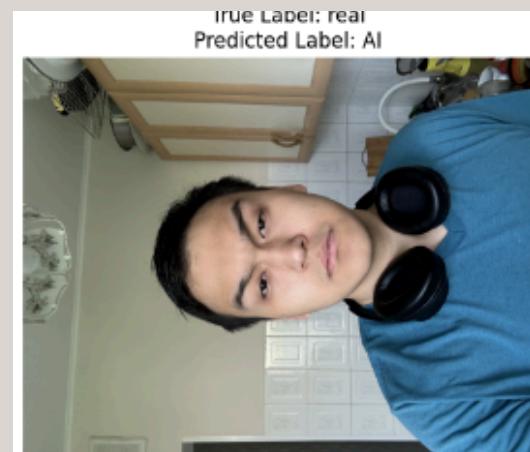
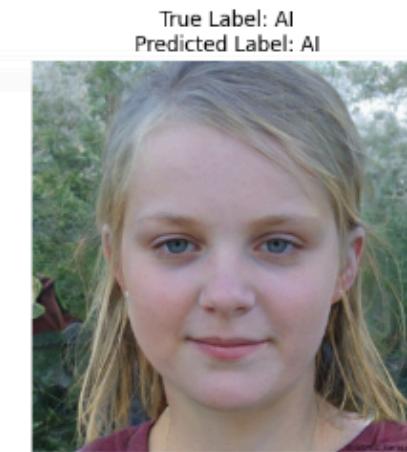
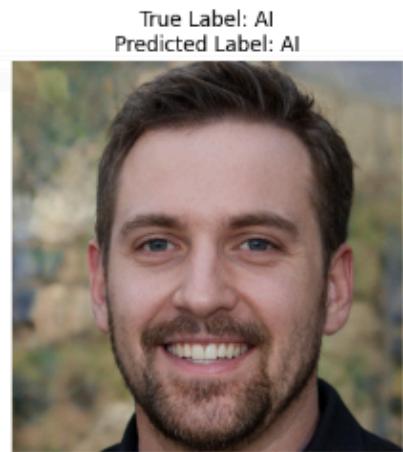


Results

ResNet101 - no freeze

Accuracy	F1-score	Precision	Recall	Val Acc
25%	0.2379	0.2408	0.2482	28%

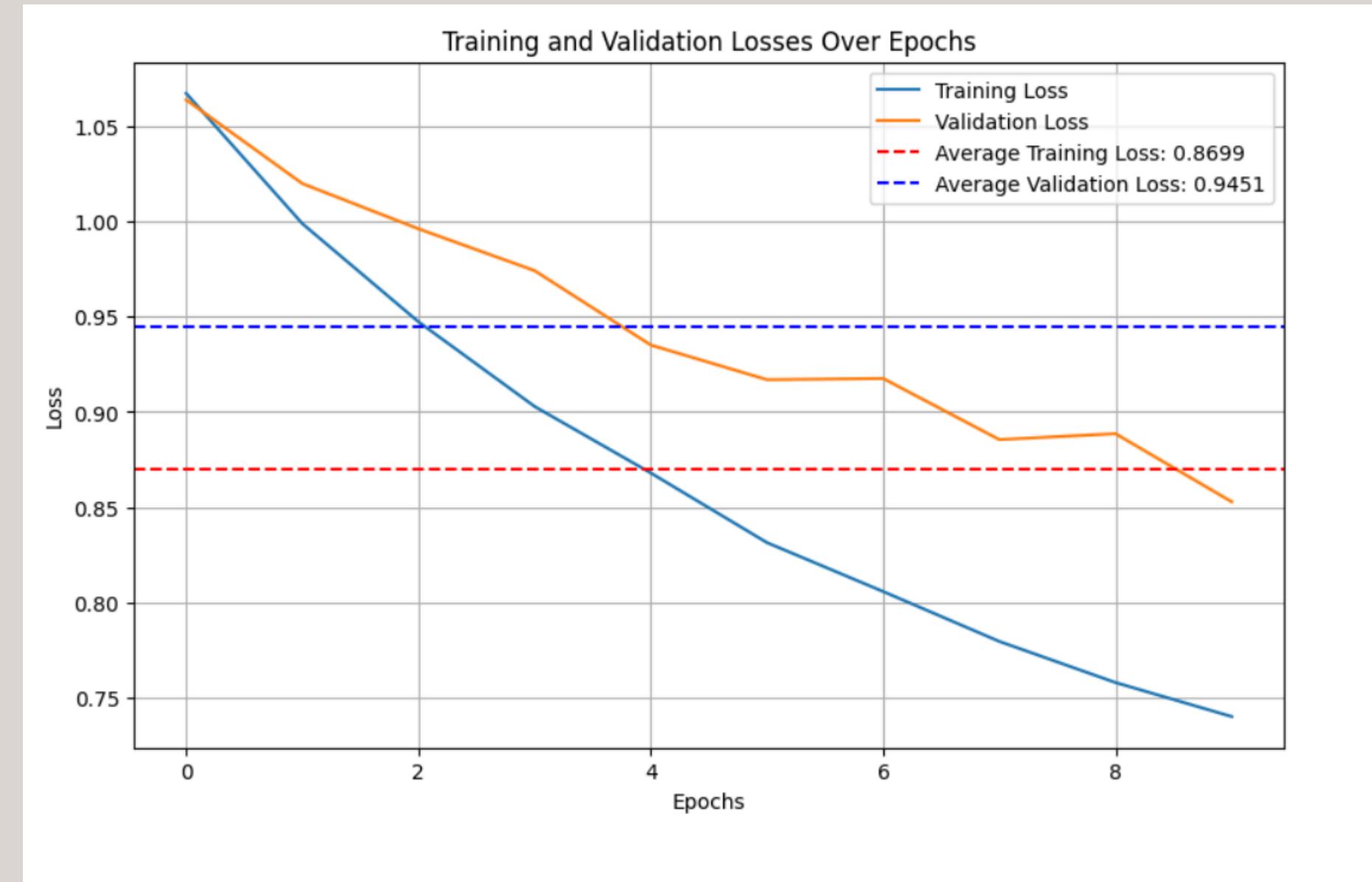
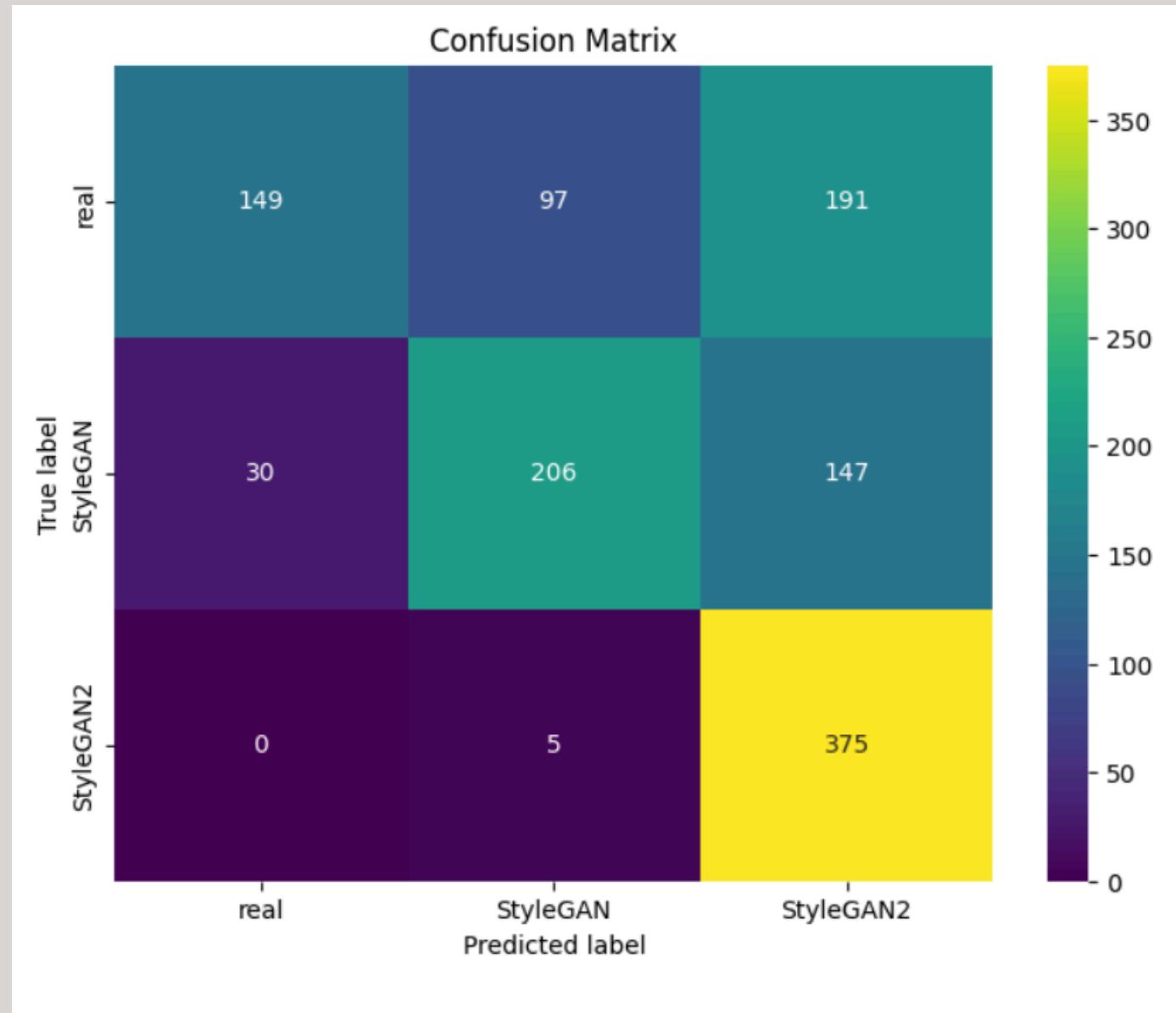


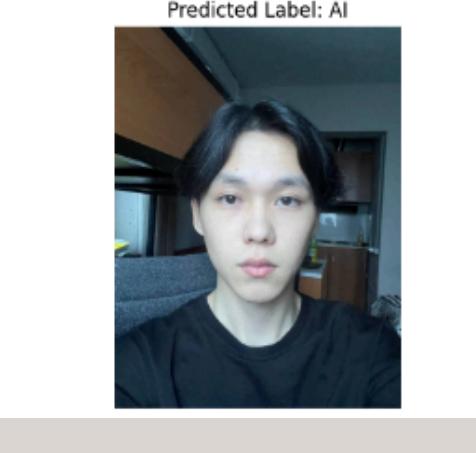
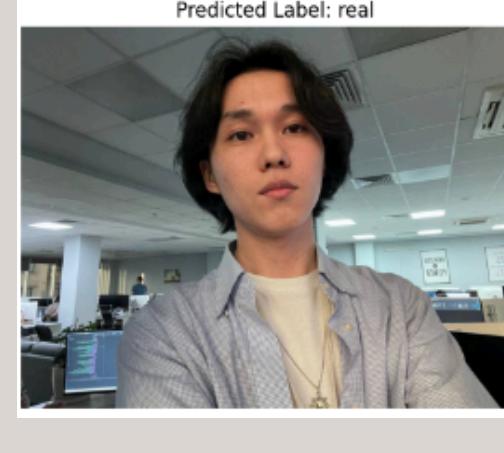
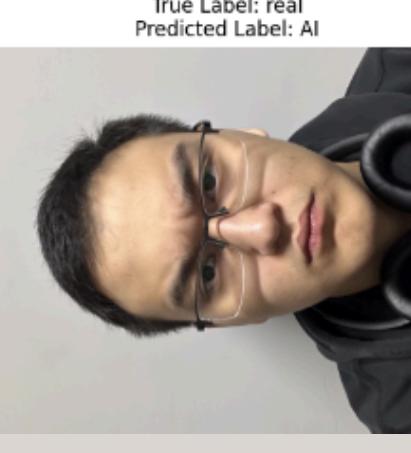
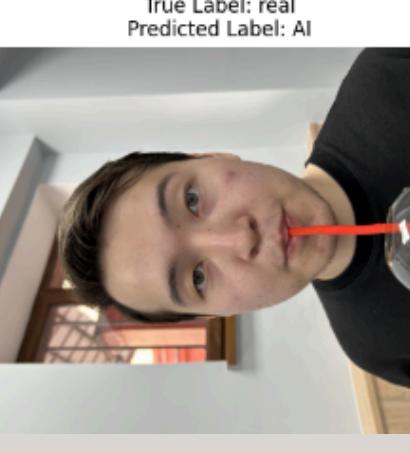
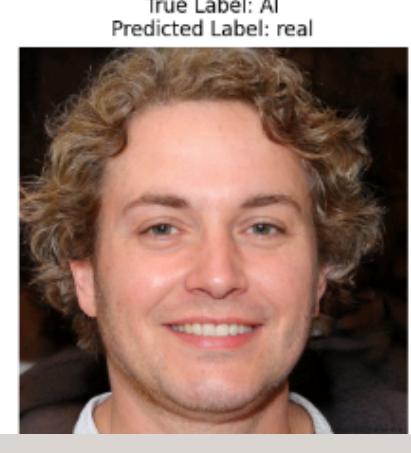
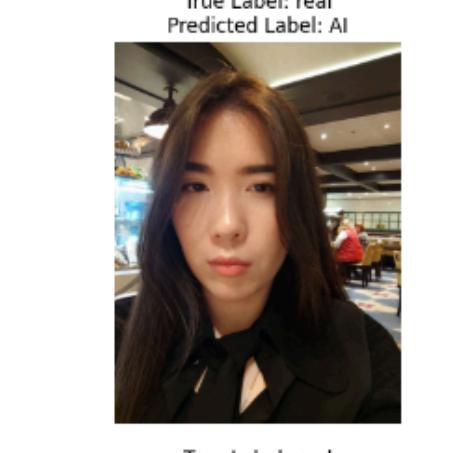
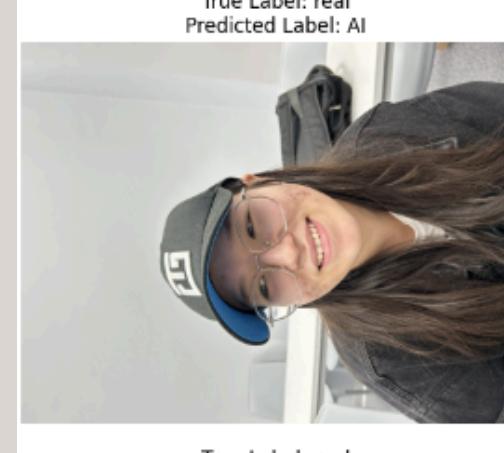
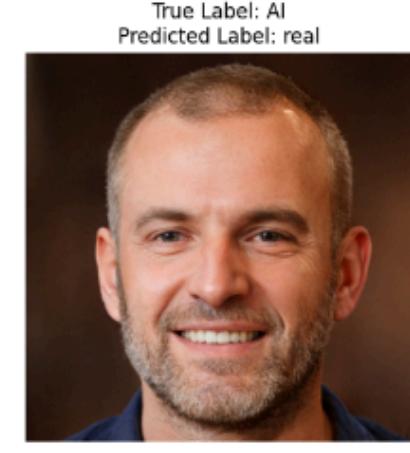
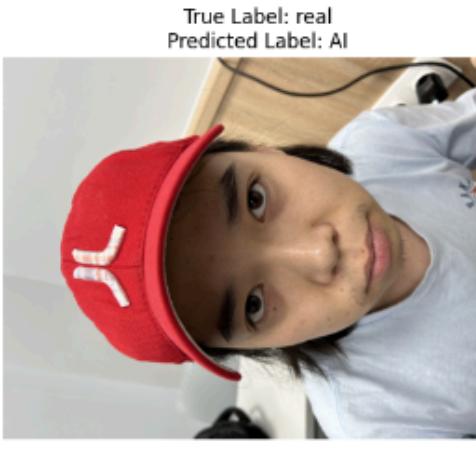
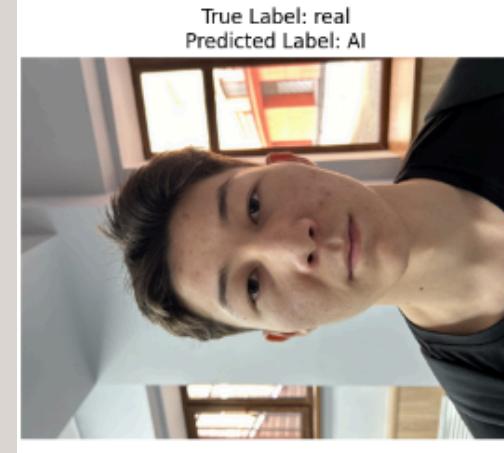
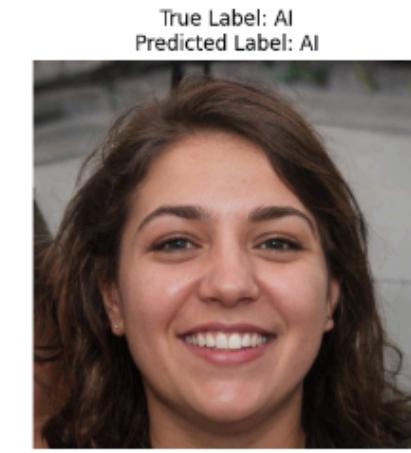
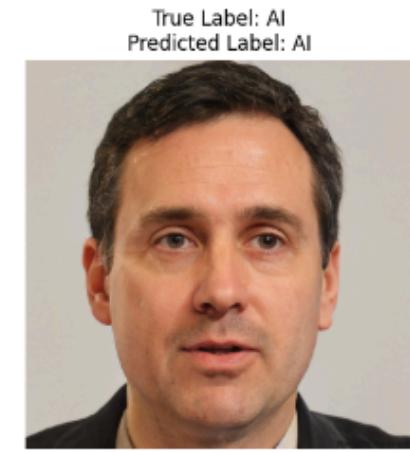
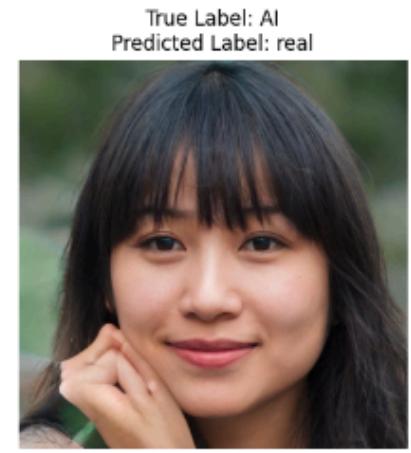
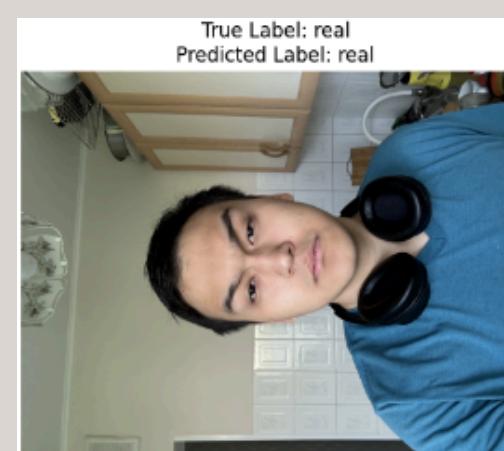
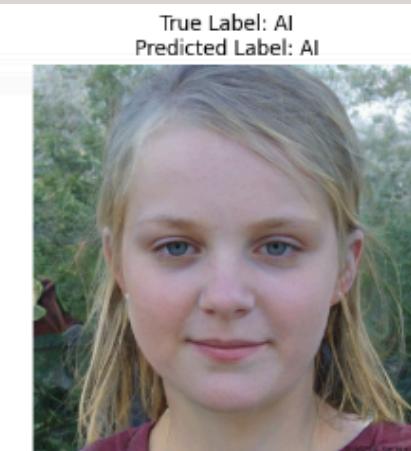
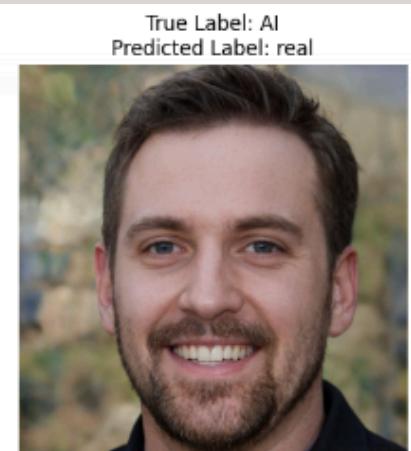


Results

ResNet101 - freeze

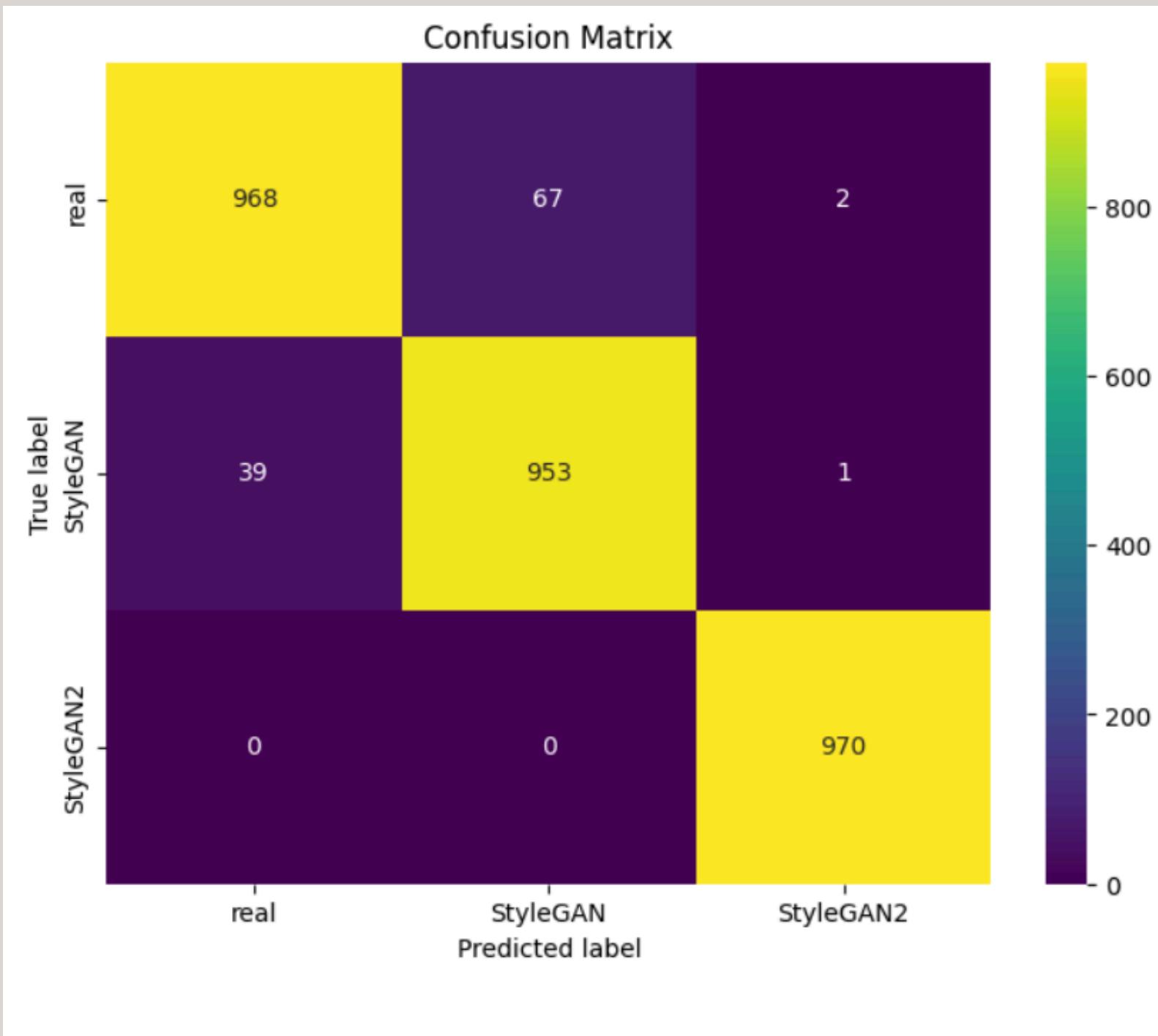
Accuracy	F1-score	Precision	Recall	Val Acc
61%	0.5887	0.6757	0.6219	60%



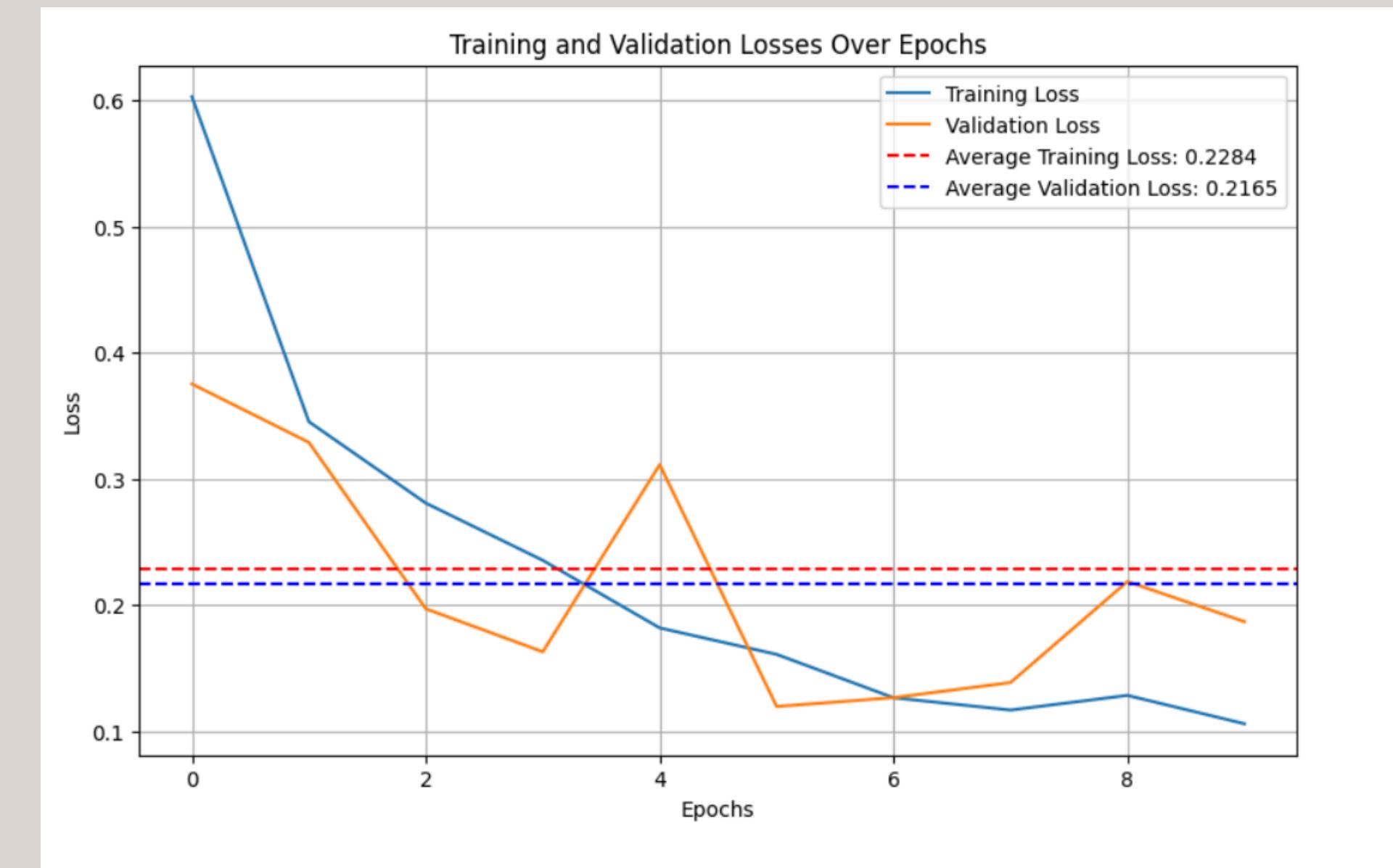


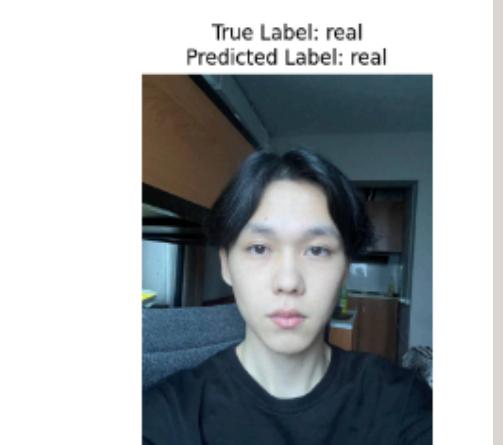
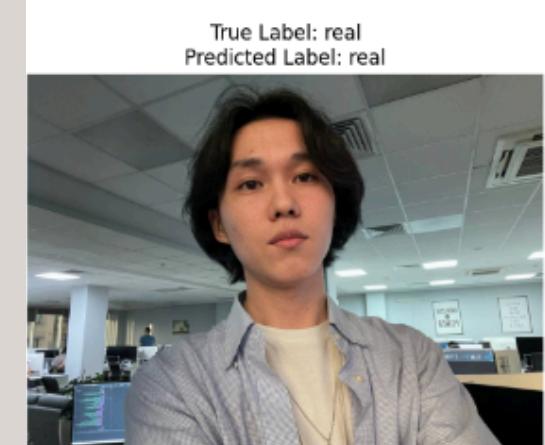
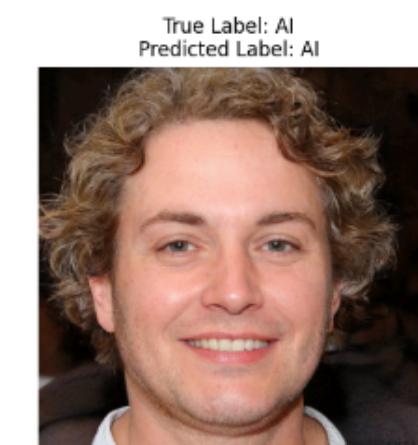
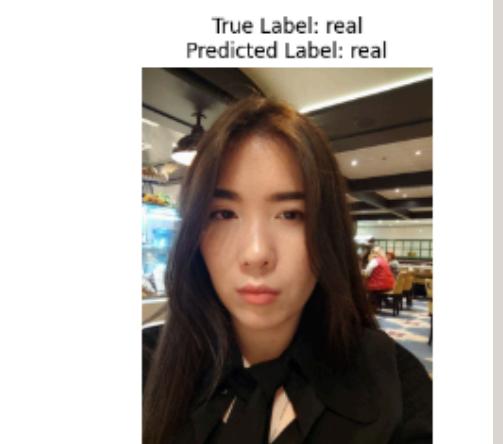
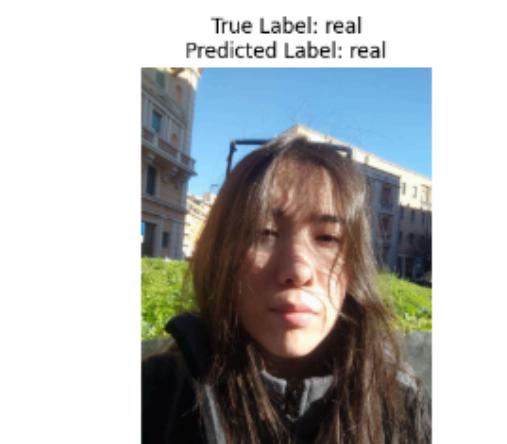
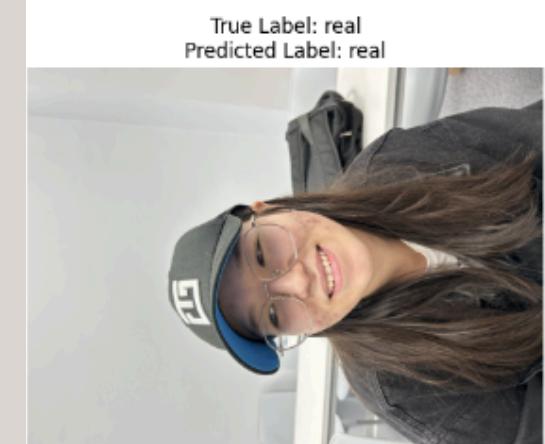
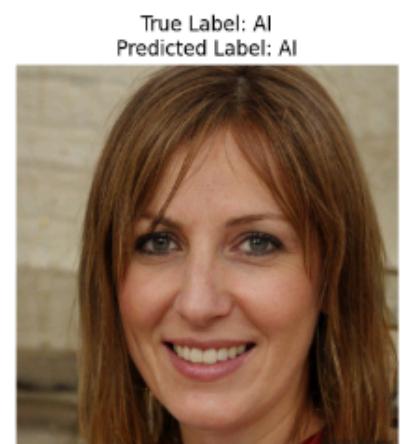
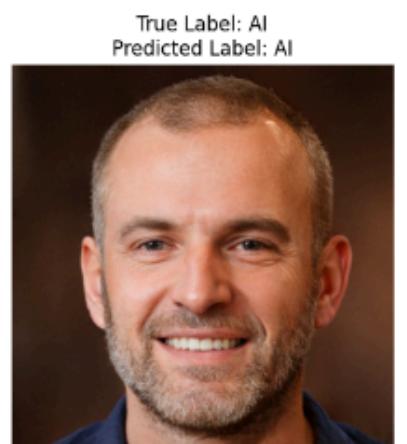
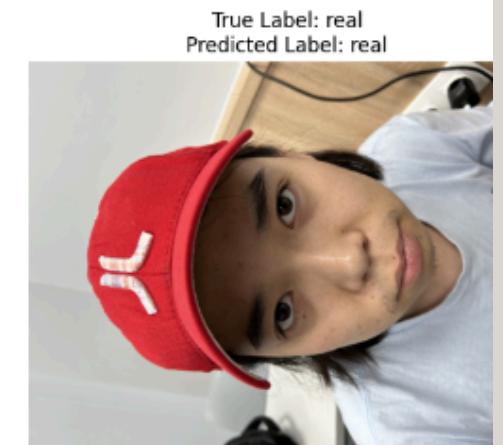
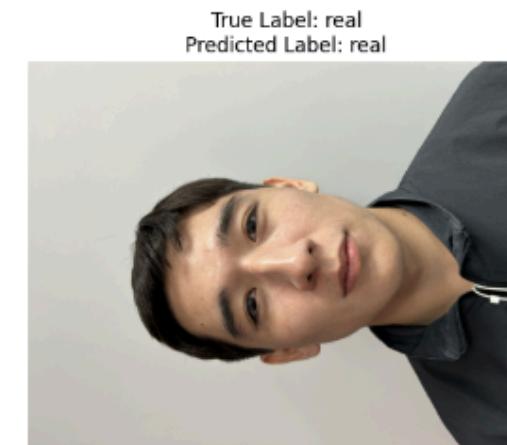
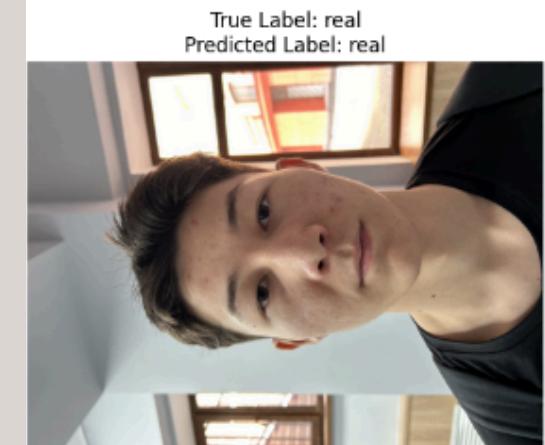
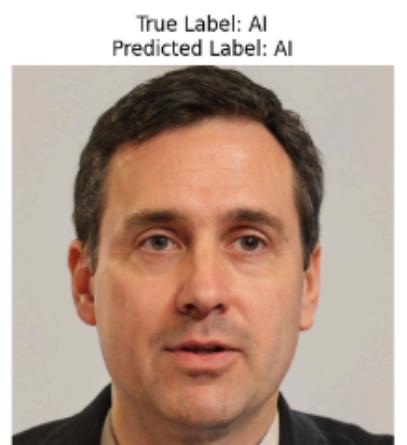
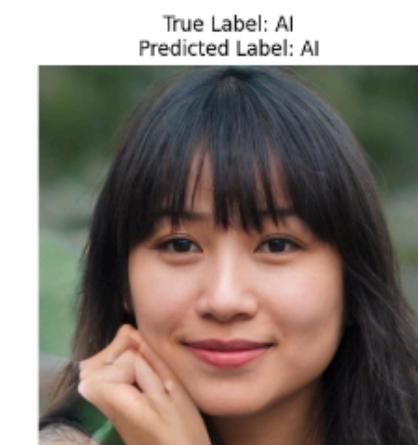
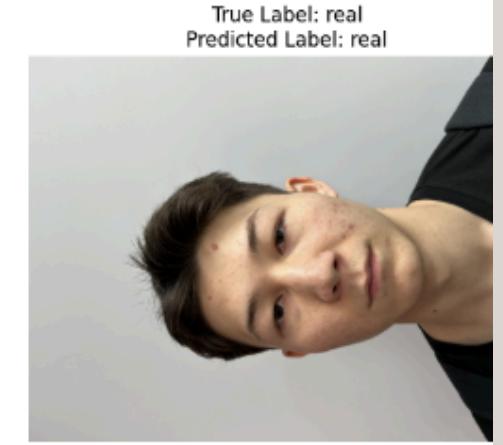
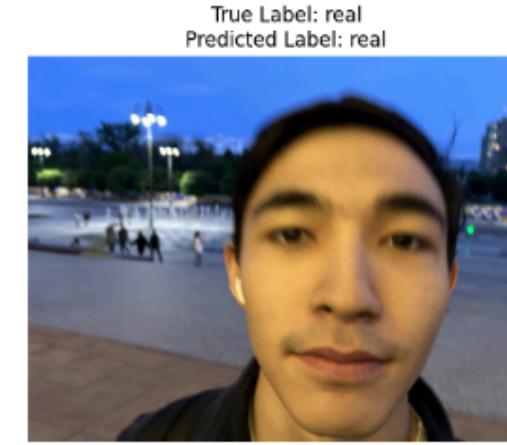
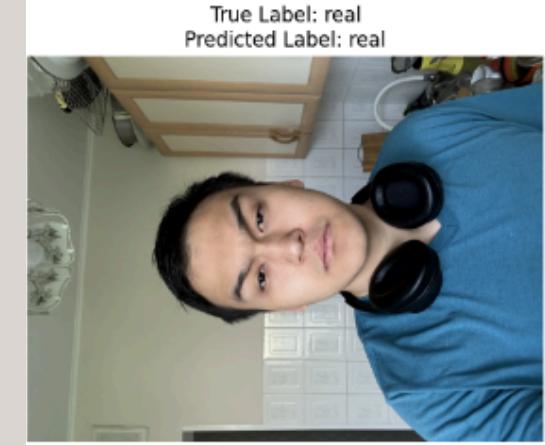
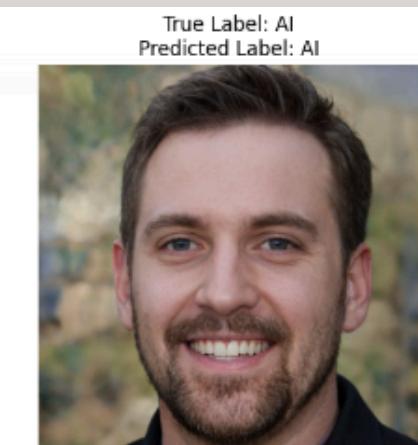
Results

VGGNet16



Accuracy	F1-score	Precision	Recall	Val acc
96%	0.9642	0.9642	0.9644	96%

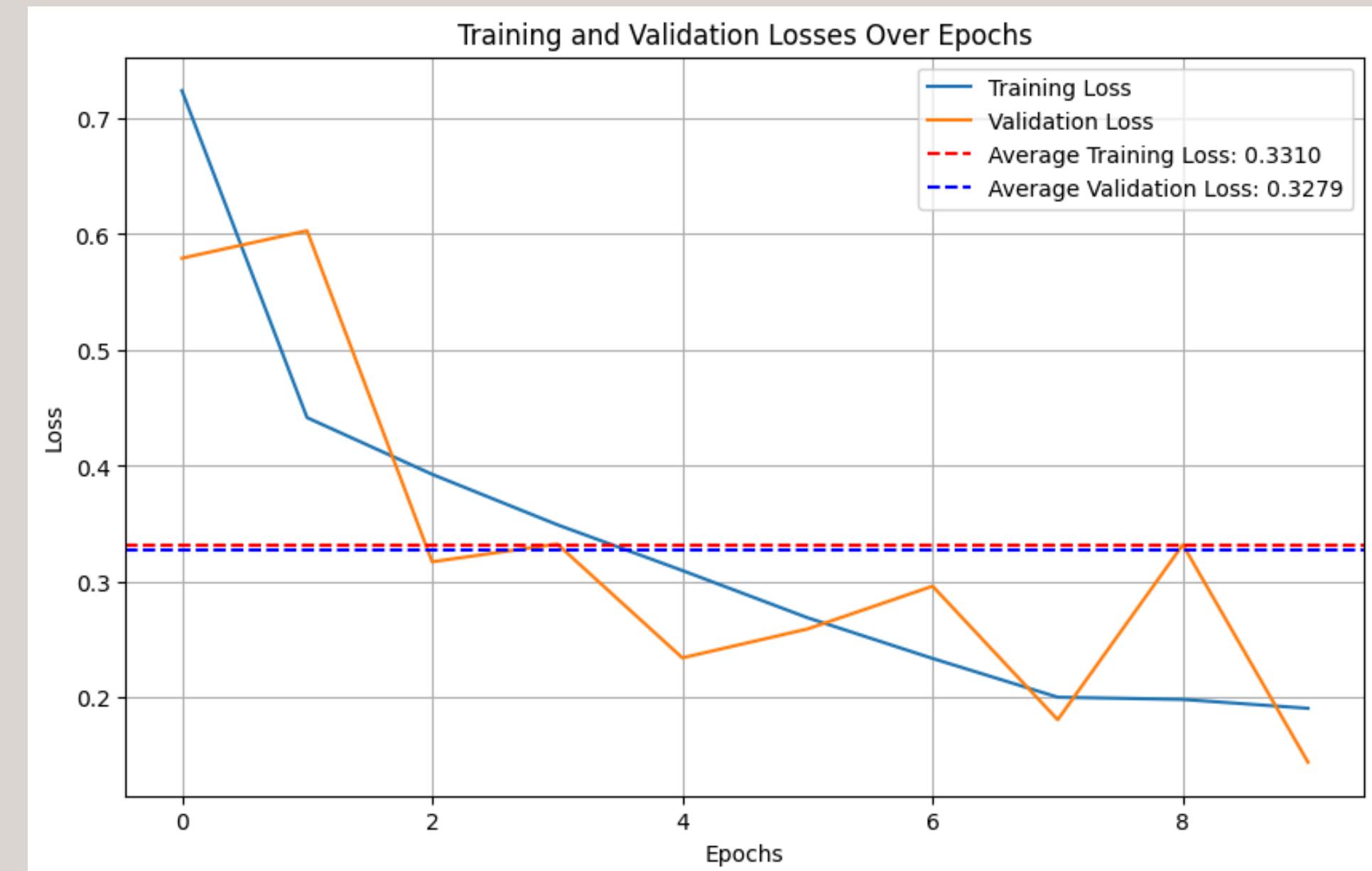
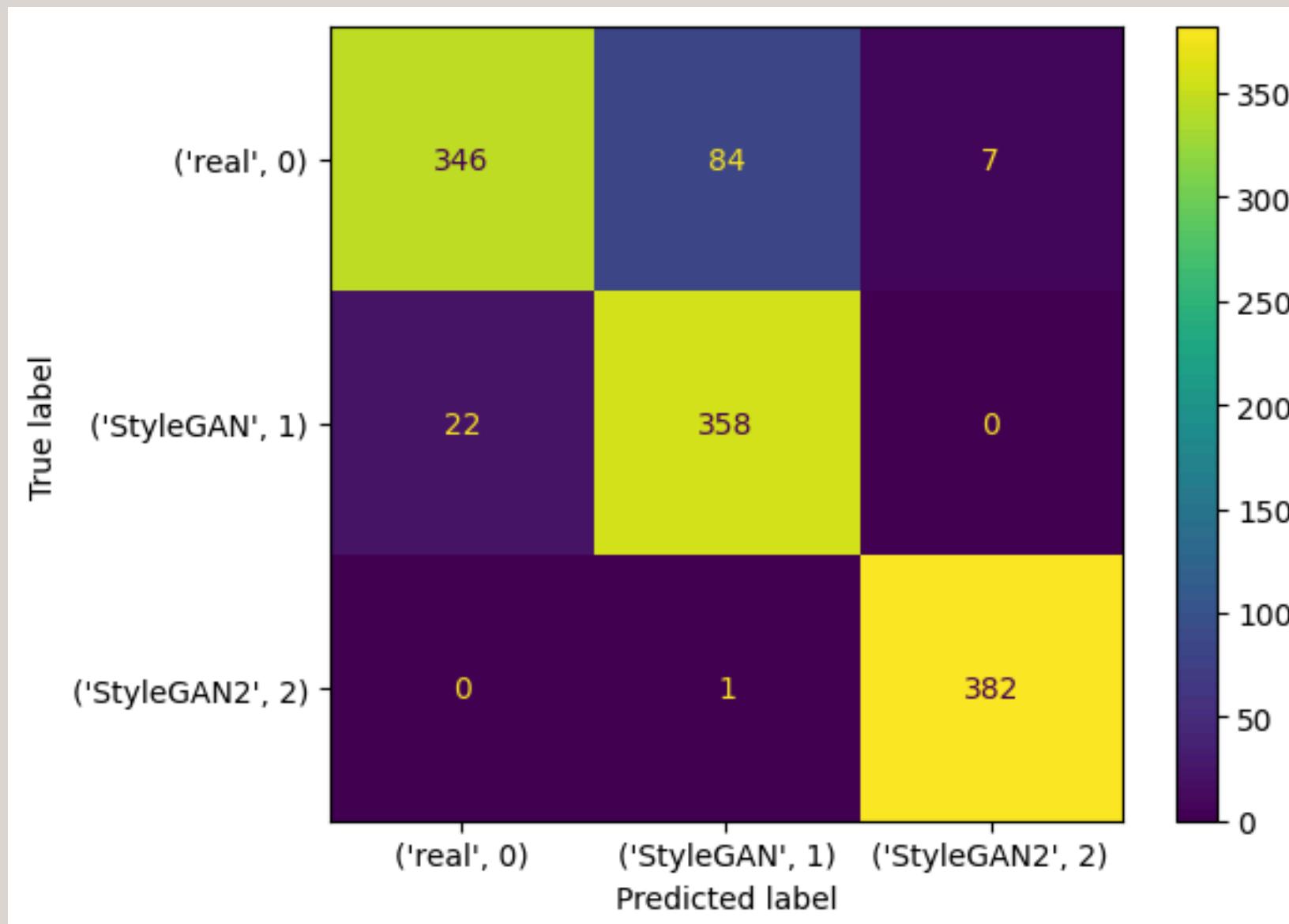


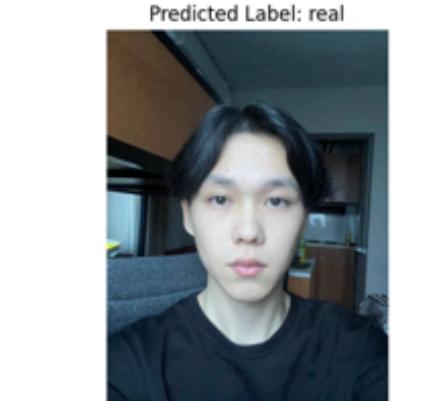
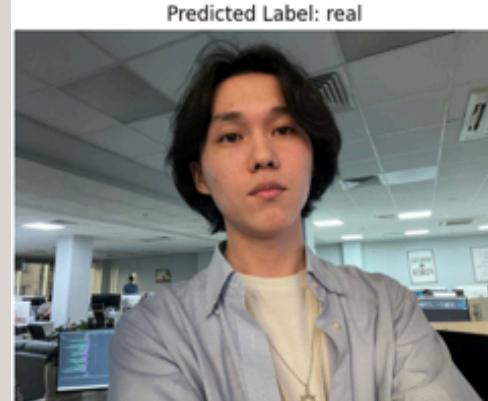
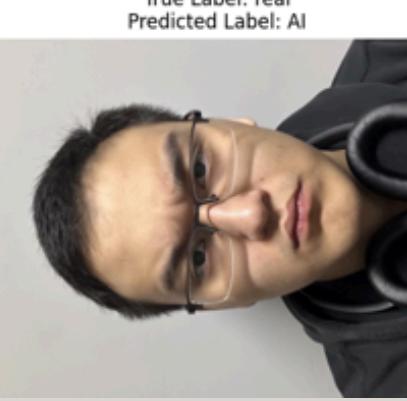
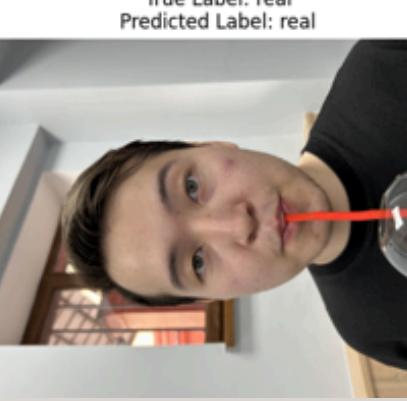
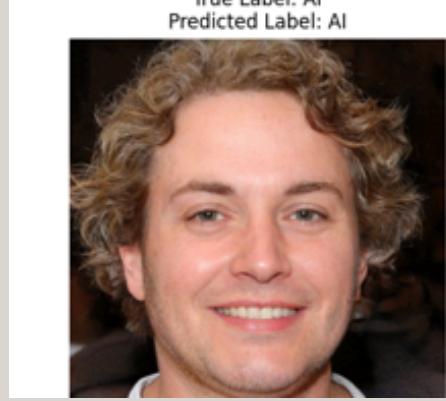
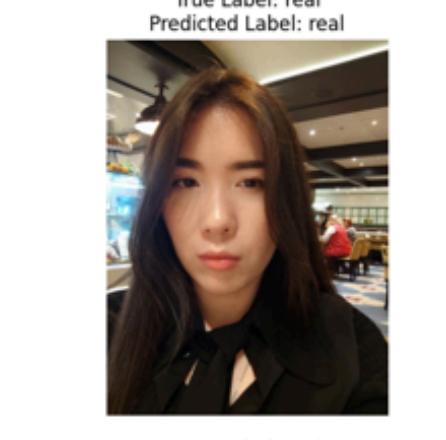
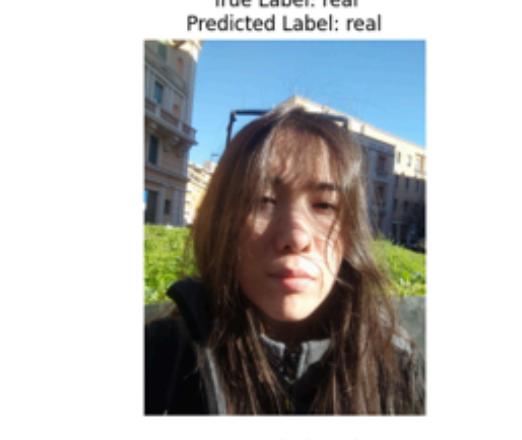
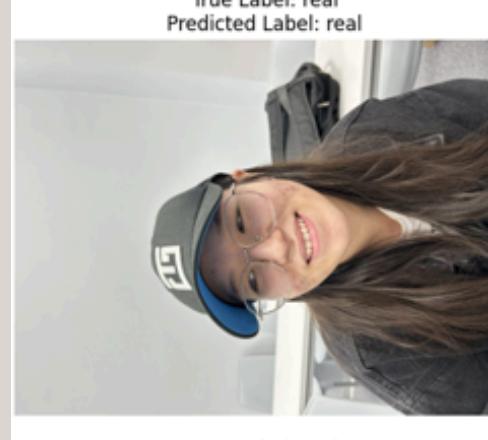
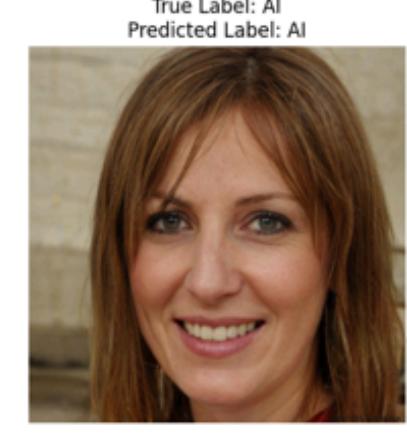
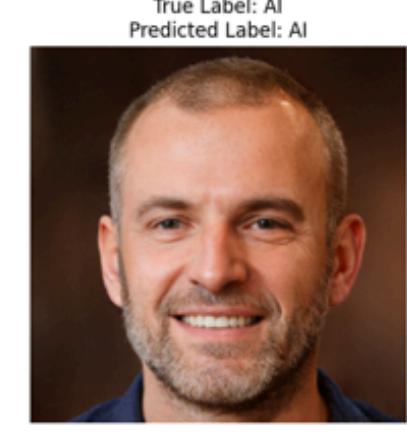
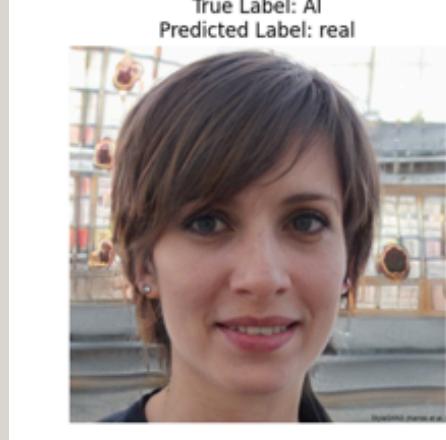
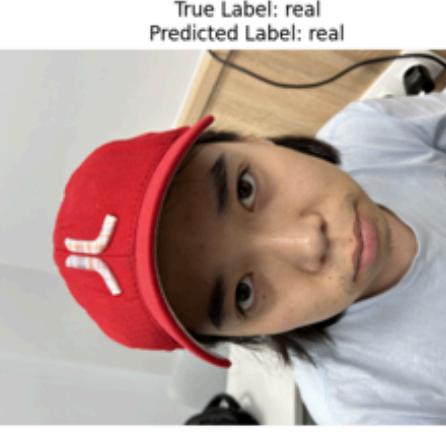
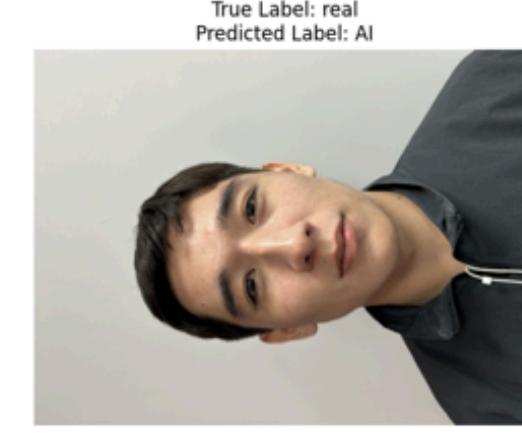
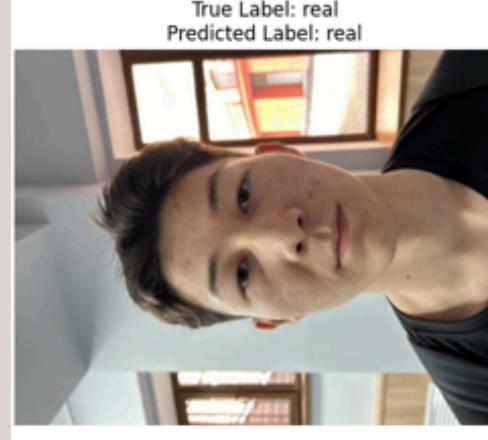
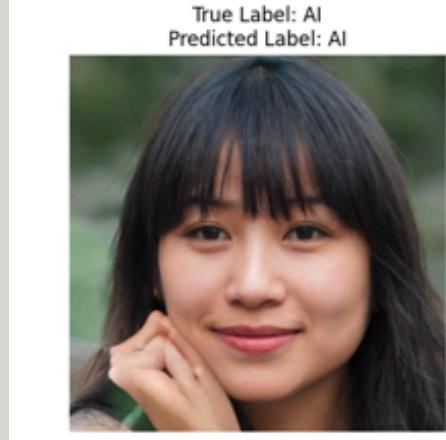
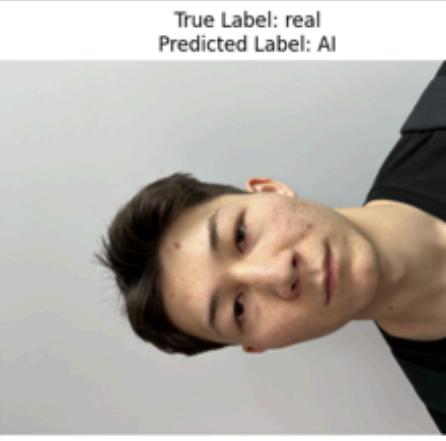
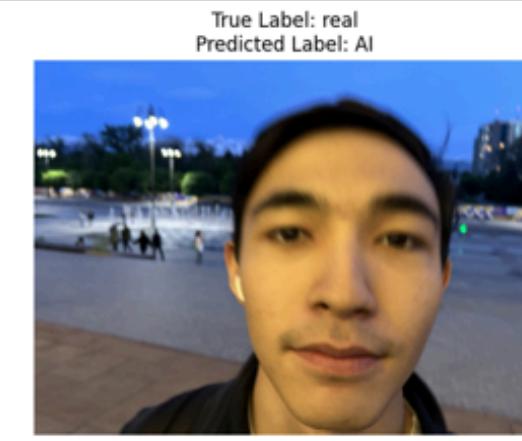
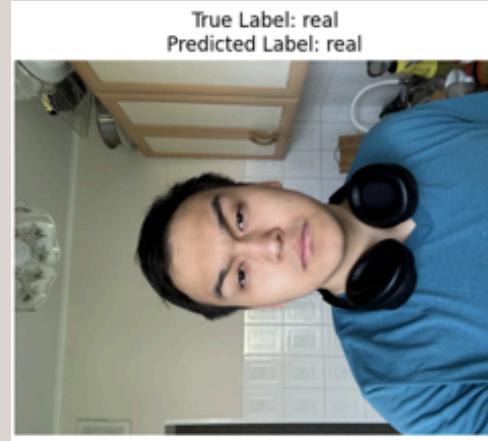
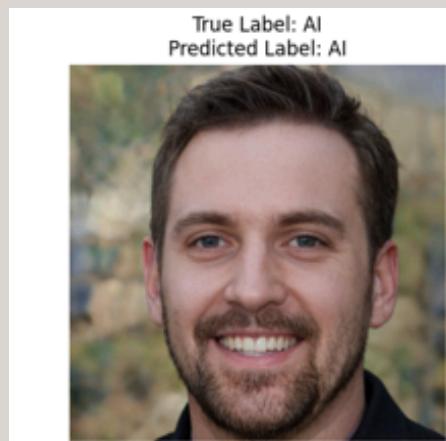


Results

VGGNet19

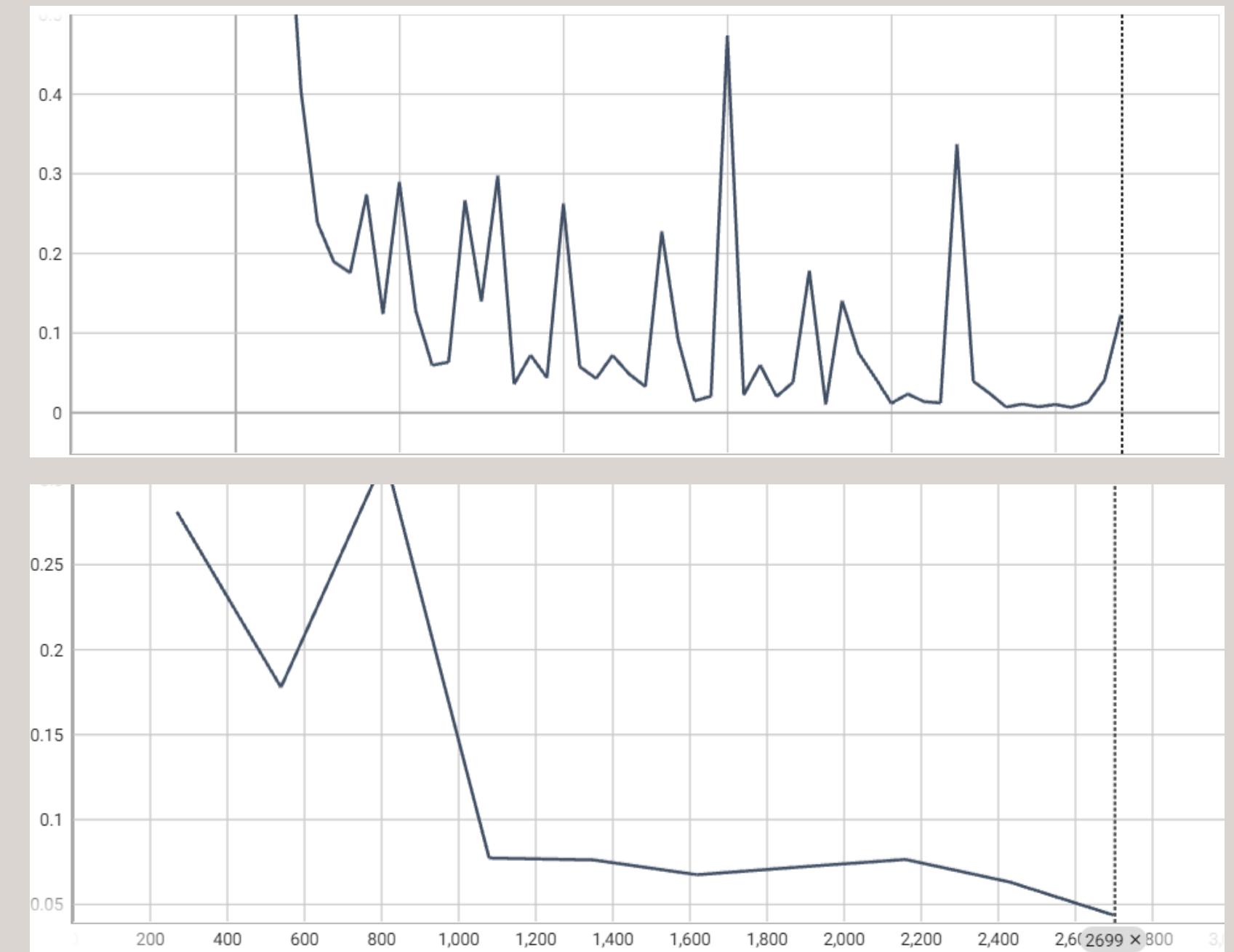
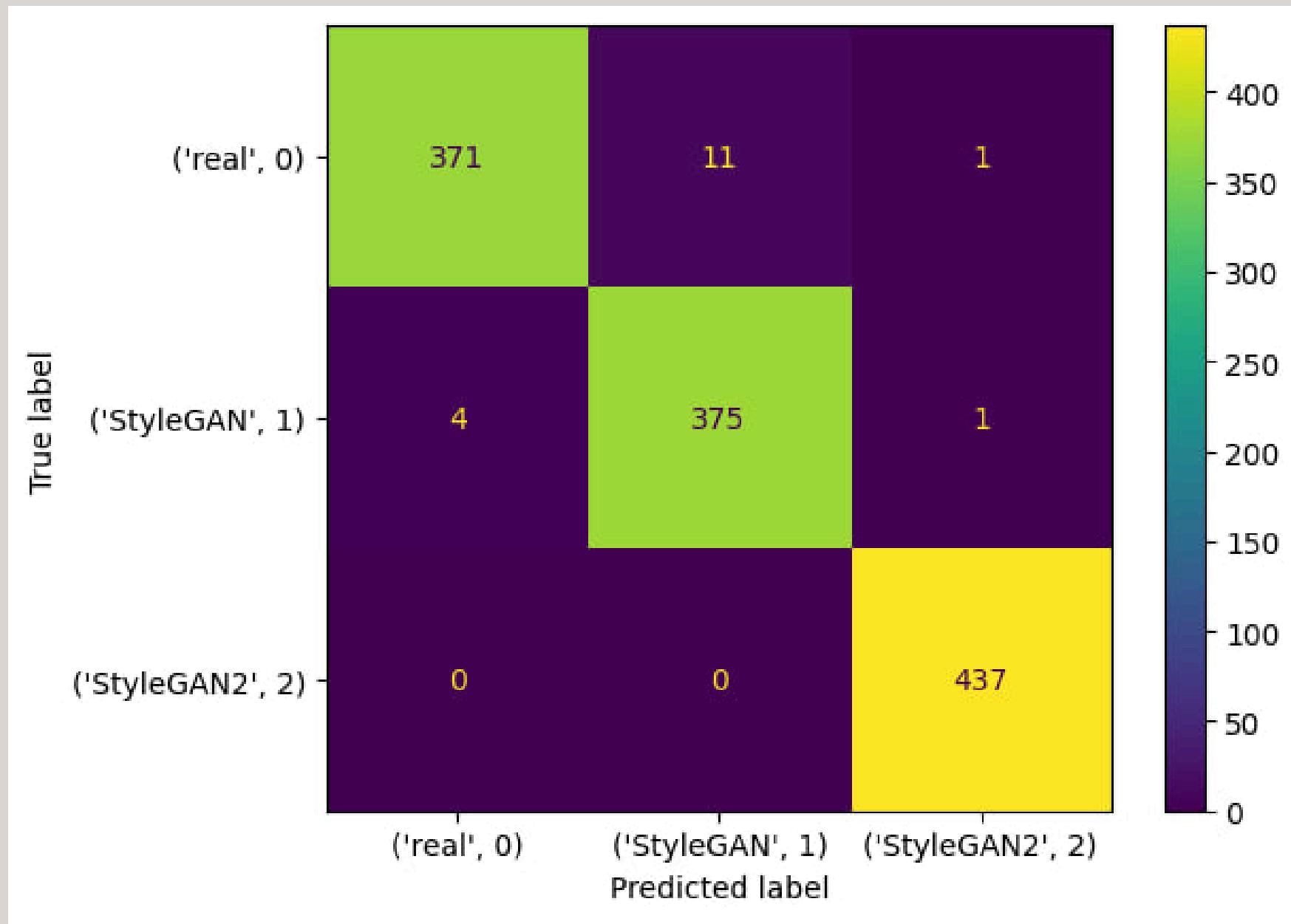
Accuracy	F1-score	Precision	Recall	Val acc
91%	0.9064	0.9101	0.9104	94%

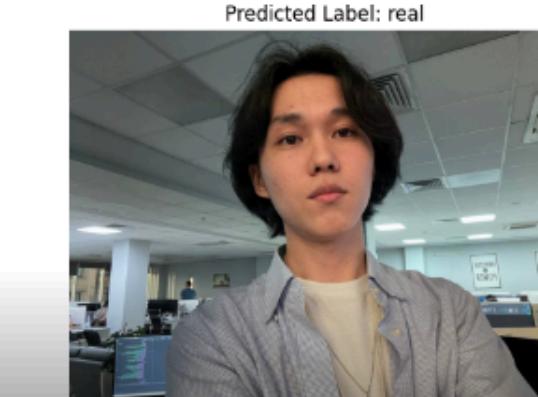
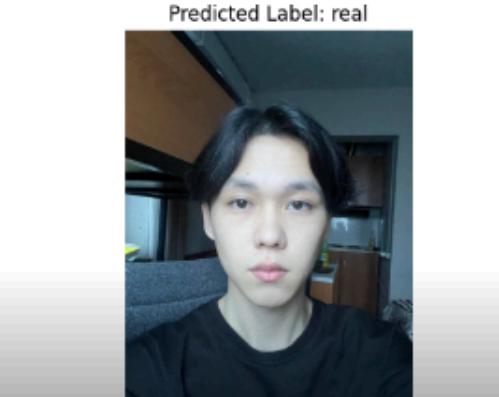
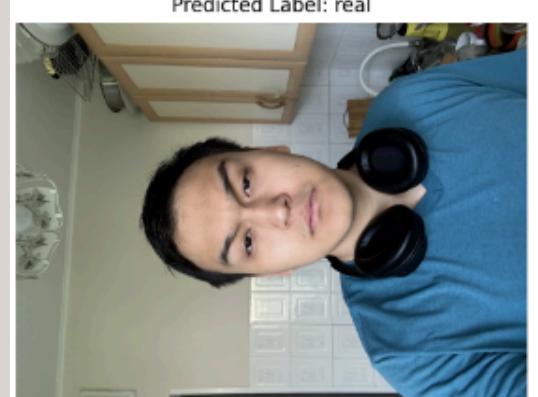
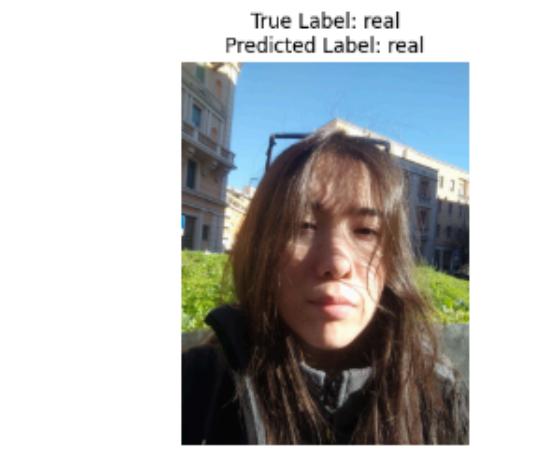
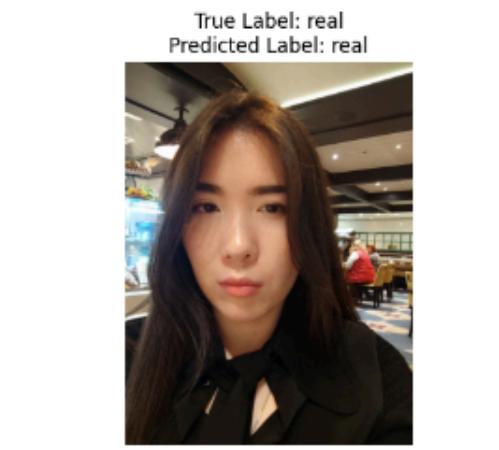
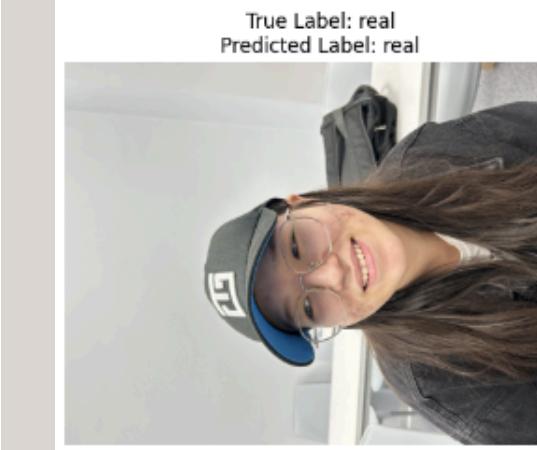
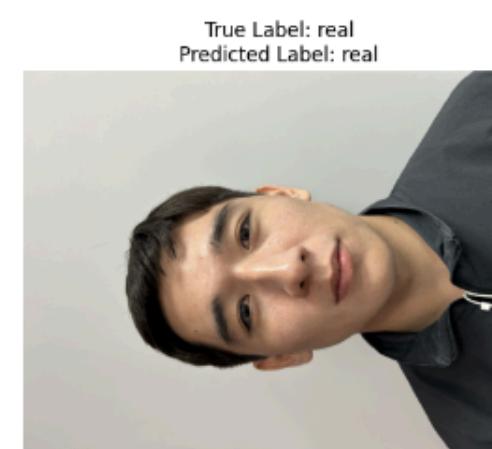
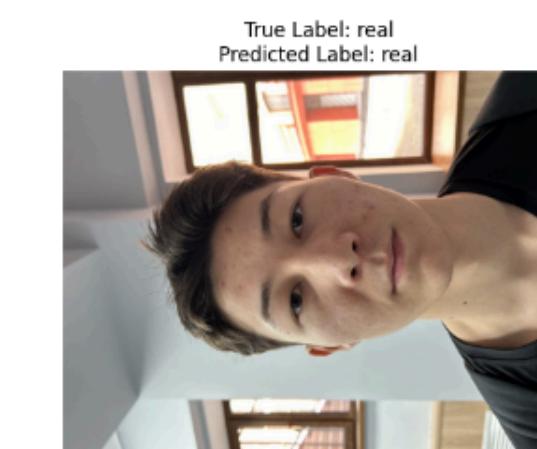
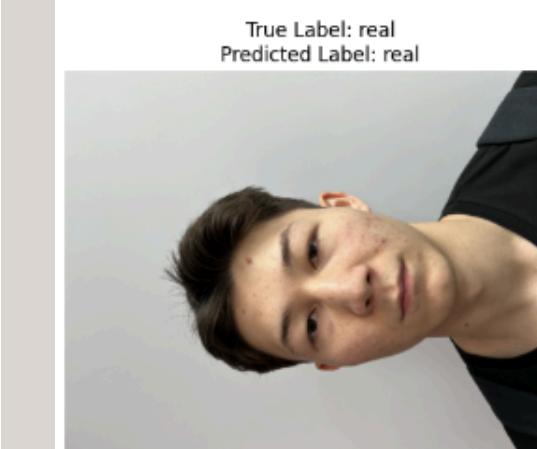
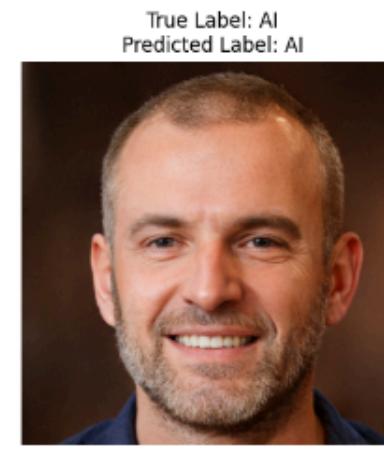
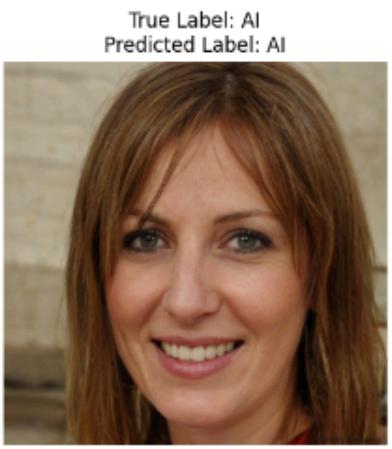
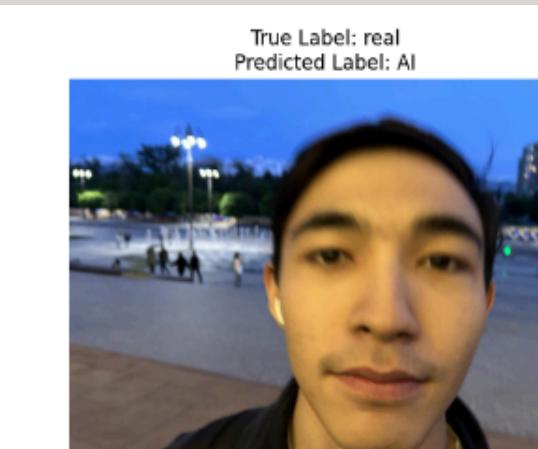
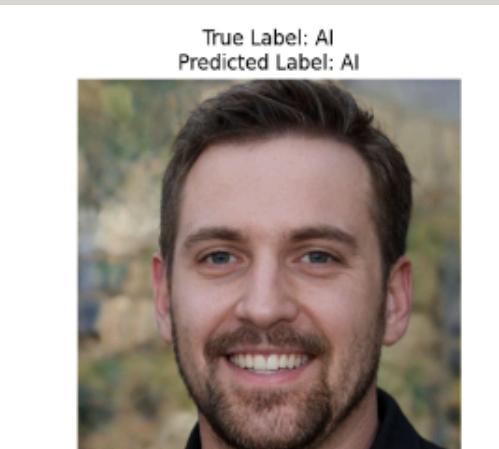
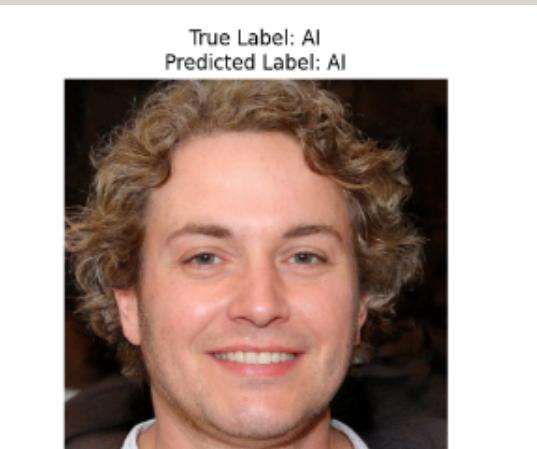
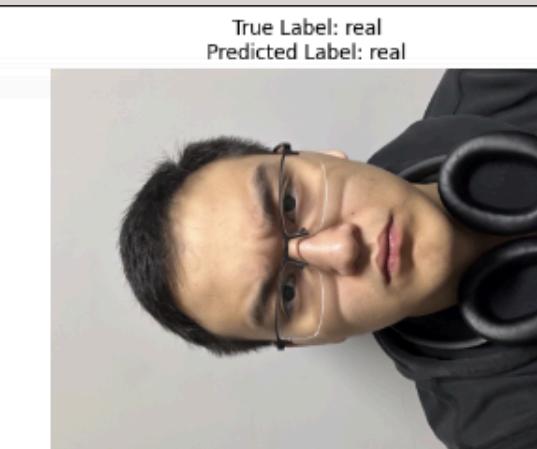
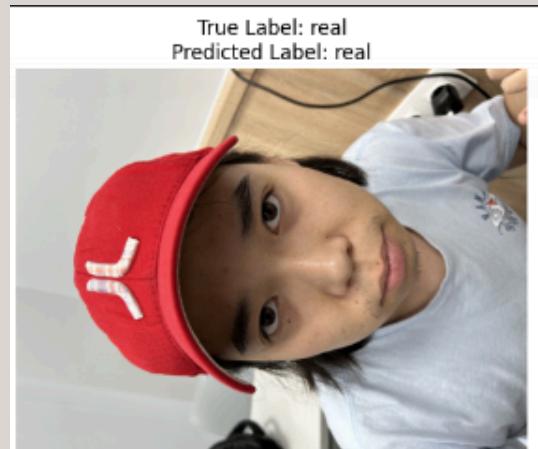




ViT

Accuracy	F1-score	Precision	Recall	Val acc
98% там вообще то 98.6% и если округлить выходит 99%	0.9852	0.9854	0.9852	98%





Overall

1. The hypothesis of article were proven again
2. We get better results for ResNet50 and VGGNet16
3. We tested ResNet101 and VGGNet19
4. We did with lesser data
5. Learned what ViT is
6. Had fun with testing model on pictures of our friends

That's all folks!