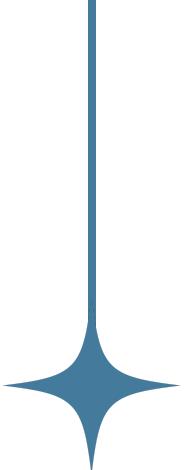


DeepLearning

Báo Cáo Đồ Án





Đề tài: Nhận diện dụng cụ lớp học





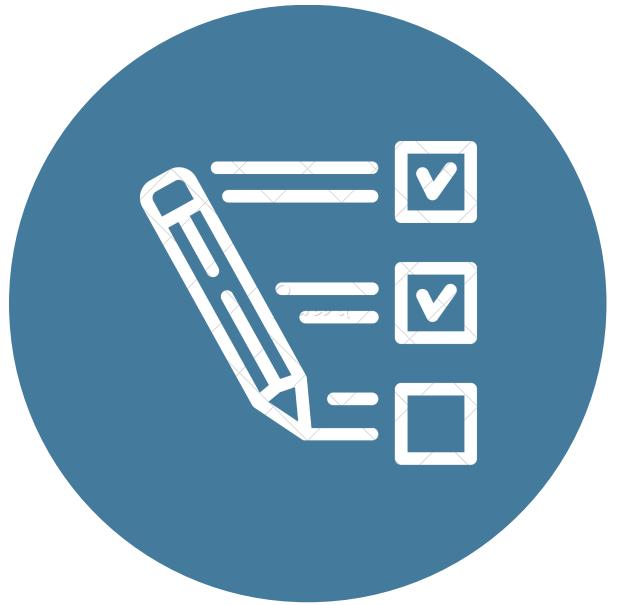
Thành viên nhóm:

- Huỳnh Tấn Kiệt - 19DH110987
- Phạm Hoàng Nhân - 19DH111002

Thư viện cần thiết



```
[ ] from tensorflow.keras.utils import plot_model
import os
from sklearn.metrics import confusion_matrix
import seaborn as sn; sn.set(font_scale=1.4)
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
import cv2
import tensorflow as tf
from tqdm import tqdm
import numpy as np
```



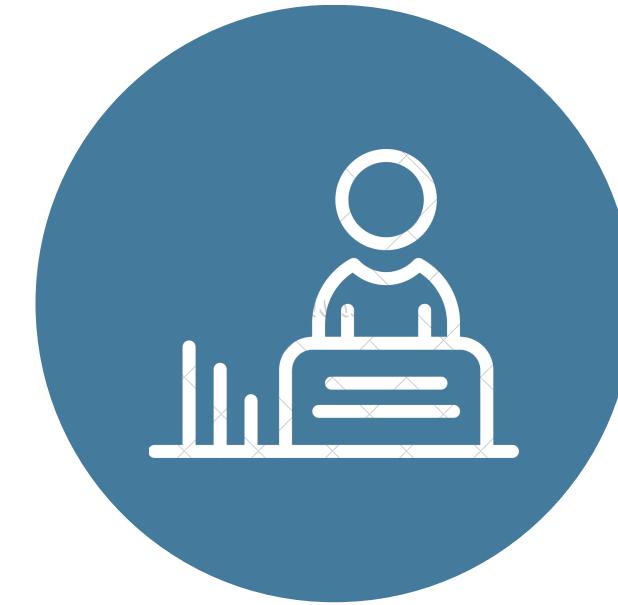
Mô hình MLP

Thử nghiệm với mô hình
MLP



Mô hình CNN

Thử nghiệm với mô hình
CNN



Pretrain

Pretrain lại mô hình



Segmentation with U-net

Dự đoán segmentation thông qua
model U-net

Gán nhãn dữ liệu

```
[ ] class_names = ['pencil', 'eraser', 'ballpoint', 'marker']
class_names_label = {class_name:i for i, class_name in enumerate(class_names)}

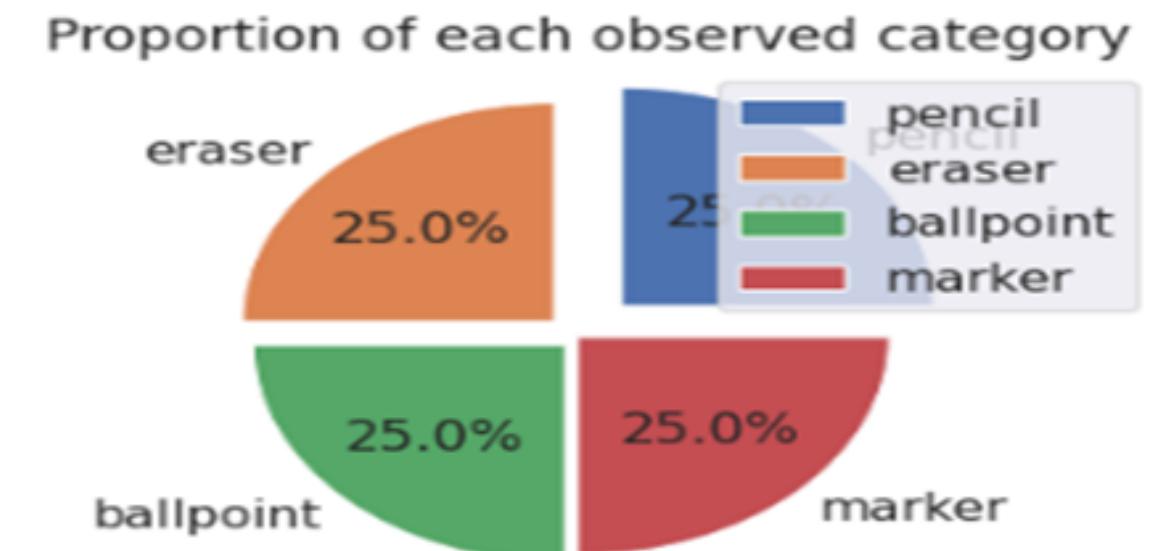
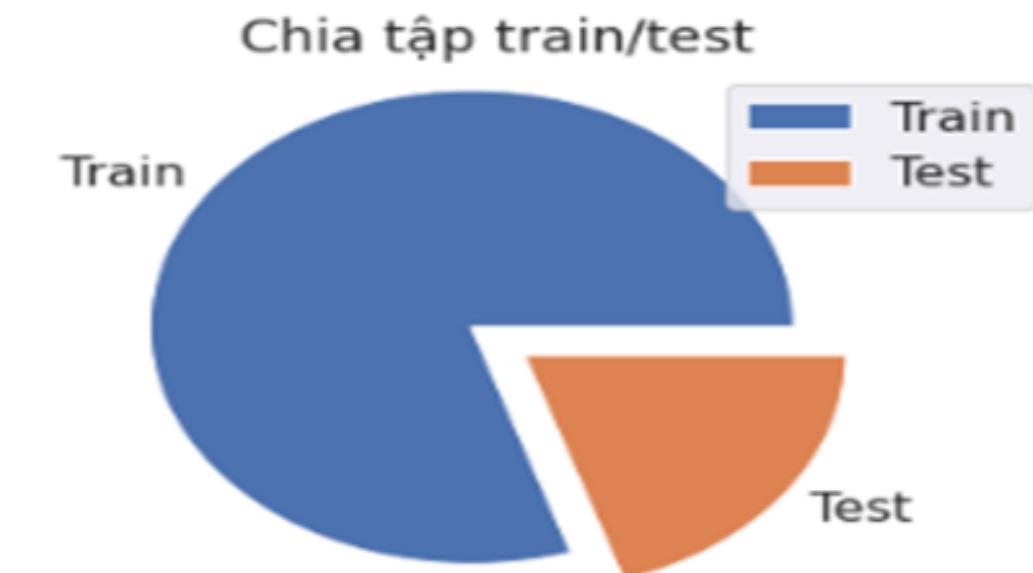
nb_classes = len(class_names)

IMAGE_SIZE = (224, 224)
```

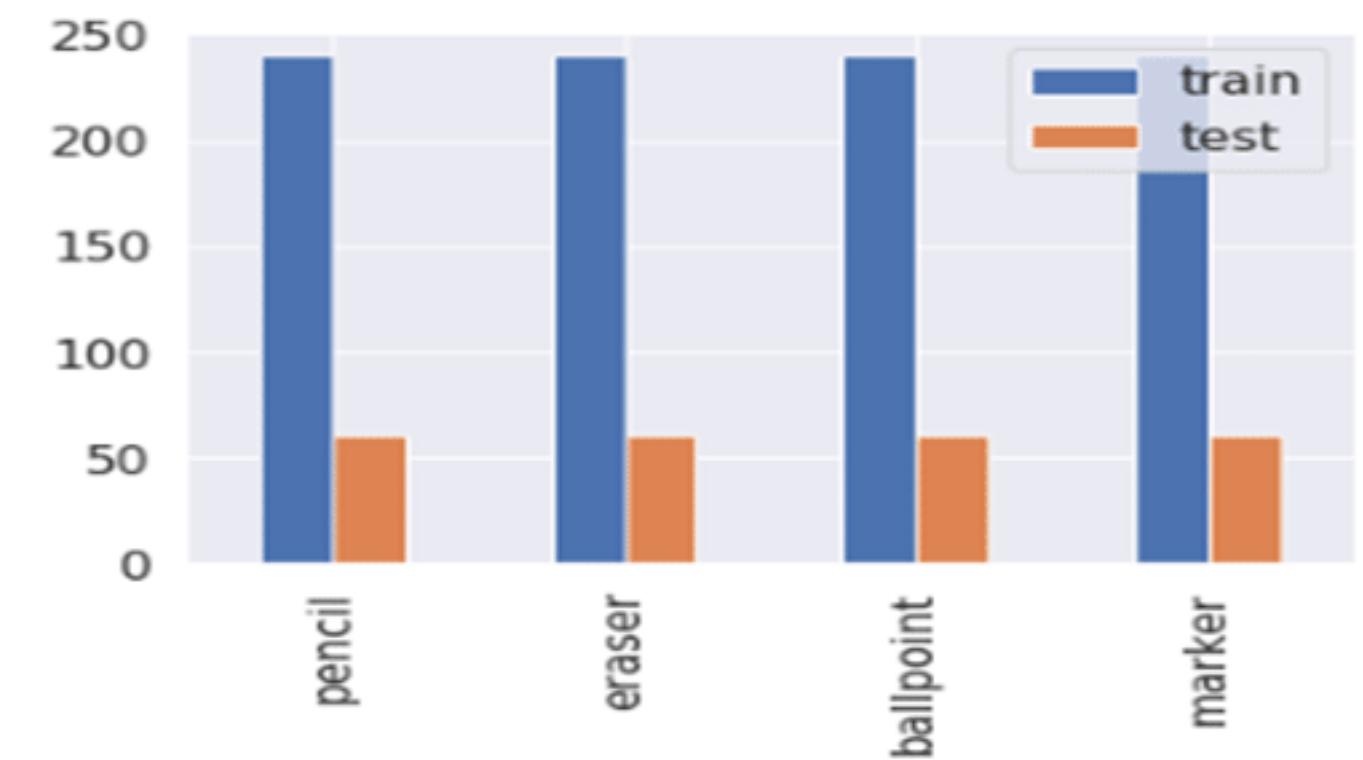




Khai báo dữ liệu và chia dữ liệu

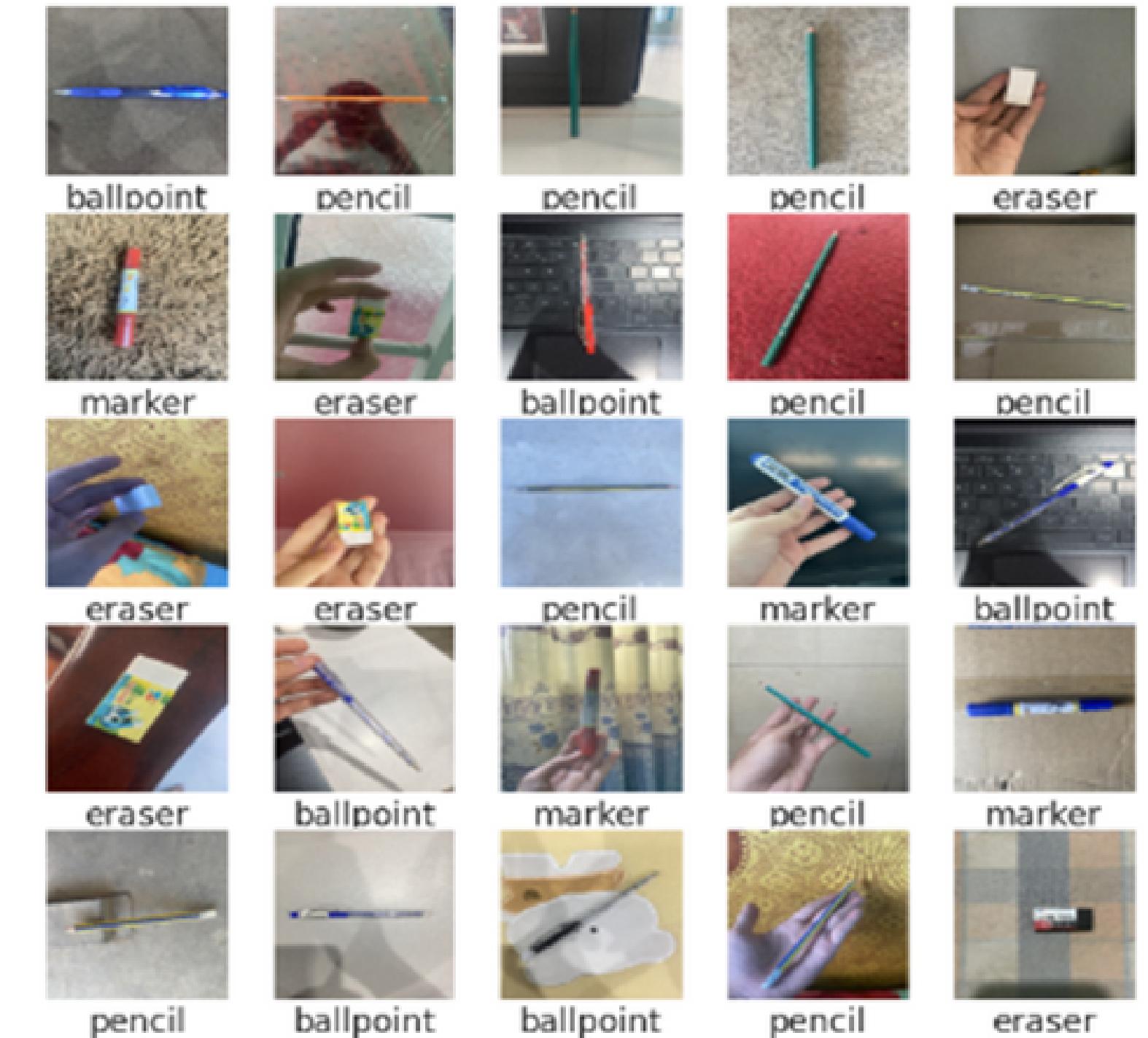


Tập dữ liệu Gồm 1200 mẫu dữ liệu:
Chia đều 300 cho các mẫu
[Pencil,ballpoint,eraser,marker]
Tỉ lệ train/test là 80/20



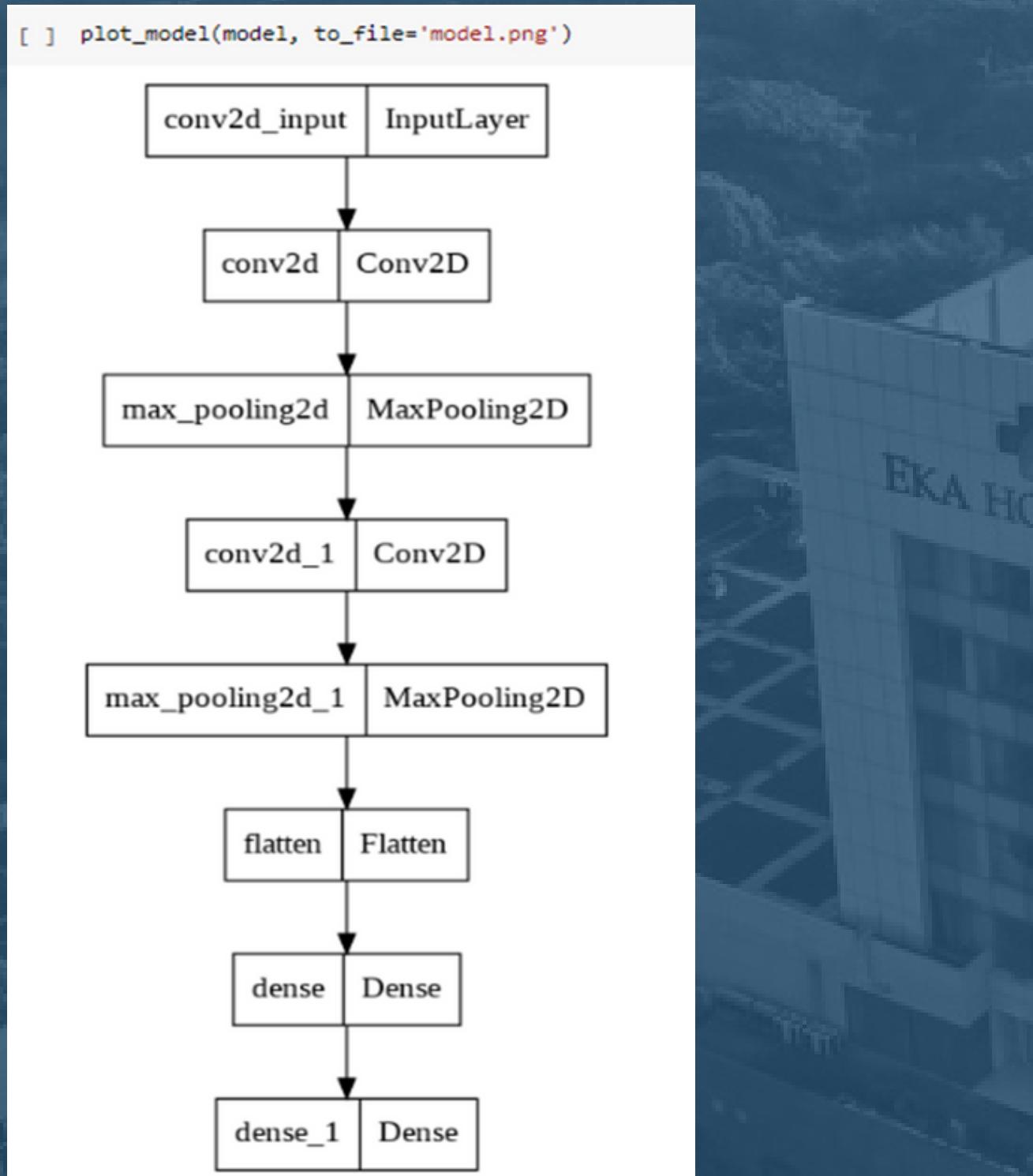
Hiển thị ảnh tương ứng và hiển thị 25 ảnh ví dụ

Image #115 : marker





CNN Model



Dự đoán của mô hình ✨

```
[ ] test_loss = model.evaluate(test_images, test_labels)
```

```
8/8 [=====] - 0s 37ms/step - loss: 1.3529 - accuracy: 0.6500
```

```
[ ] predictions = model.predict(test_images)      # Vector of probabilities
pred_labels = np.argmax(predictions, axis = 1) # We take the highest probability
```

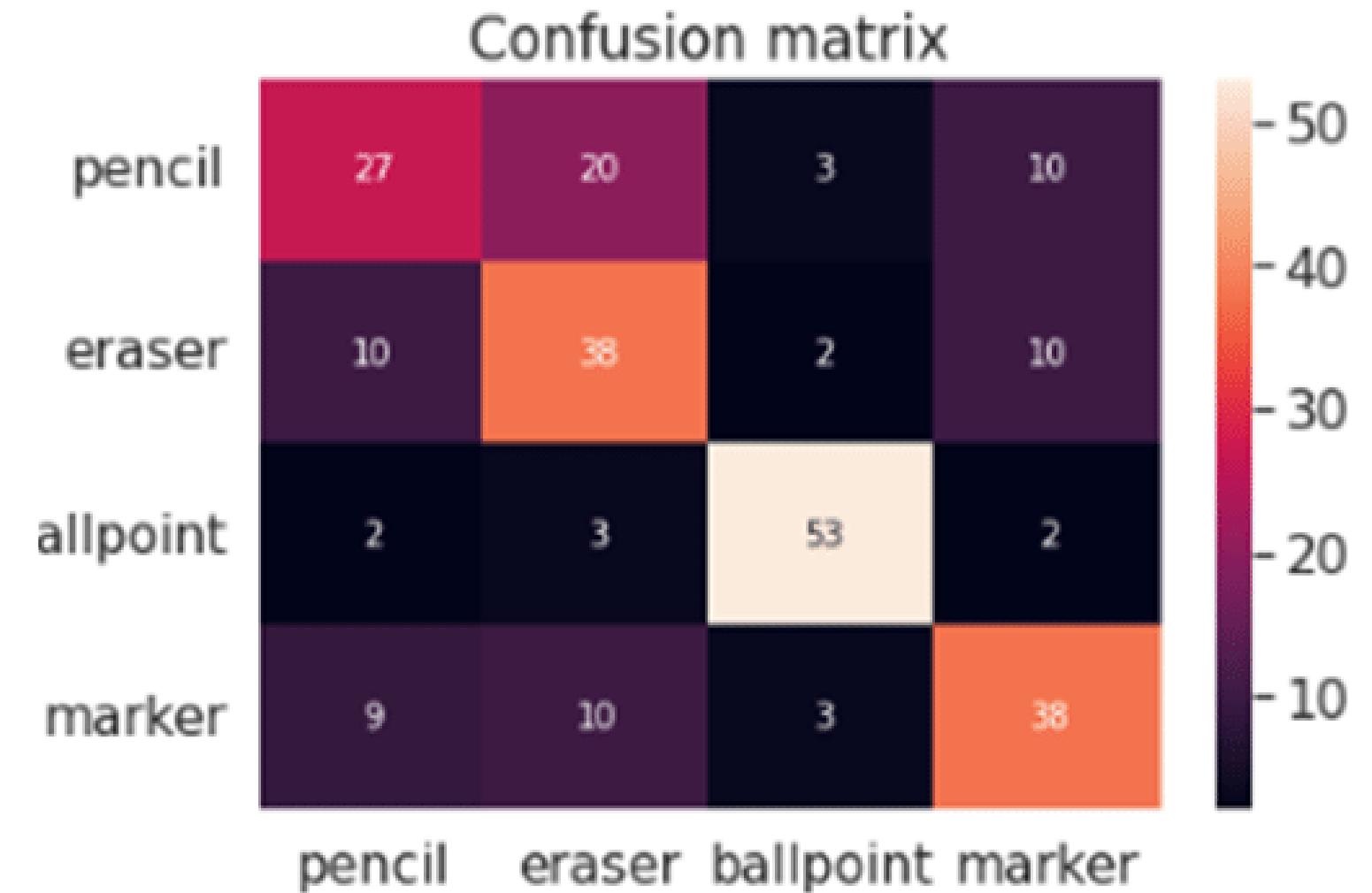
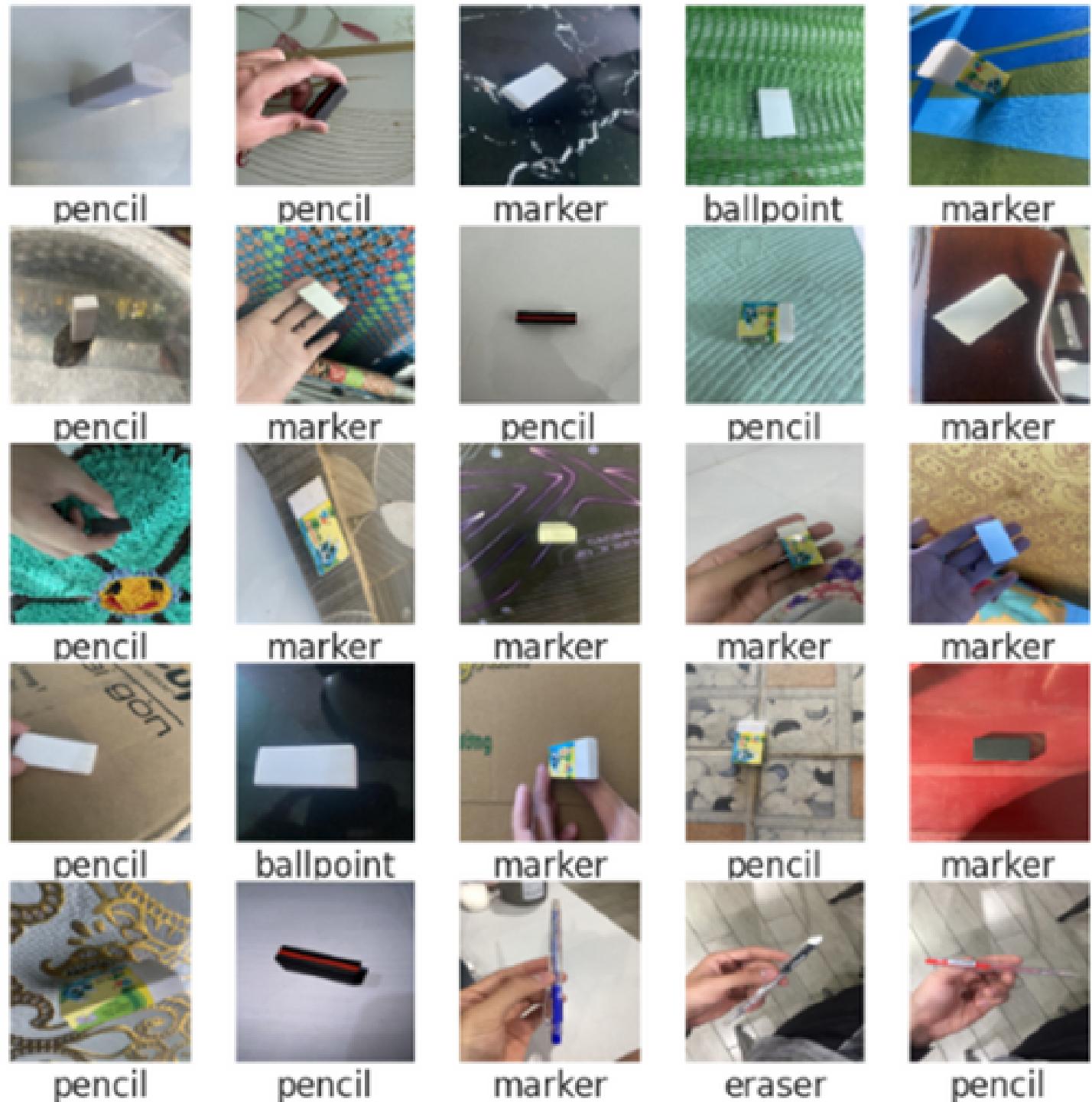
```
display_random_image(class_names, test_images, pred_labels)
```

```
8/8 [=====] - 0s 18ms/step
Image #54 : eraser
```



	image_size = 224,224	channels = 3									
	batch_size	epochs	optimizer	loss	metrics	validation_split	accuracy	val_accuracy	loss	val_loss	predictions
CNN	128	20	adam	categorical_crossentropy	accuracy	20%	0.51	0.52	1.17	1.09	0.40
MLP	128	30	adam	sparse_categorical_crossentropy	accuracy	20%	1.00	0.72	0.01	1.11	0.75

Hiển thị 25 ảnh bị gán nhãn sai - Ma Trận Nhầm Lẫn



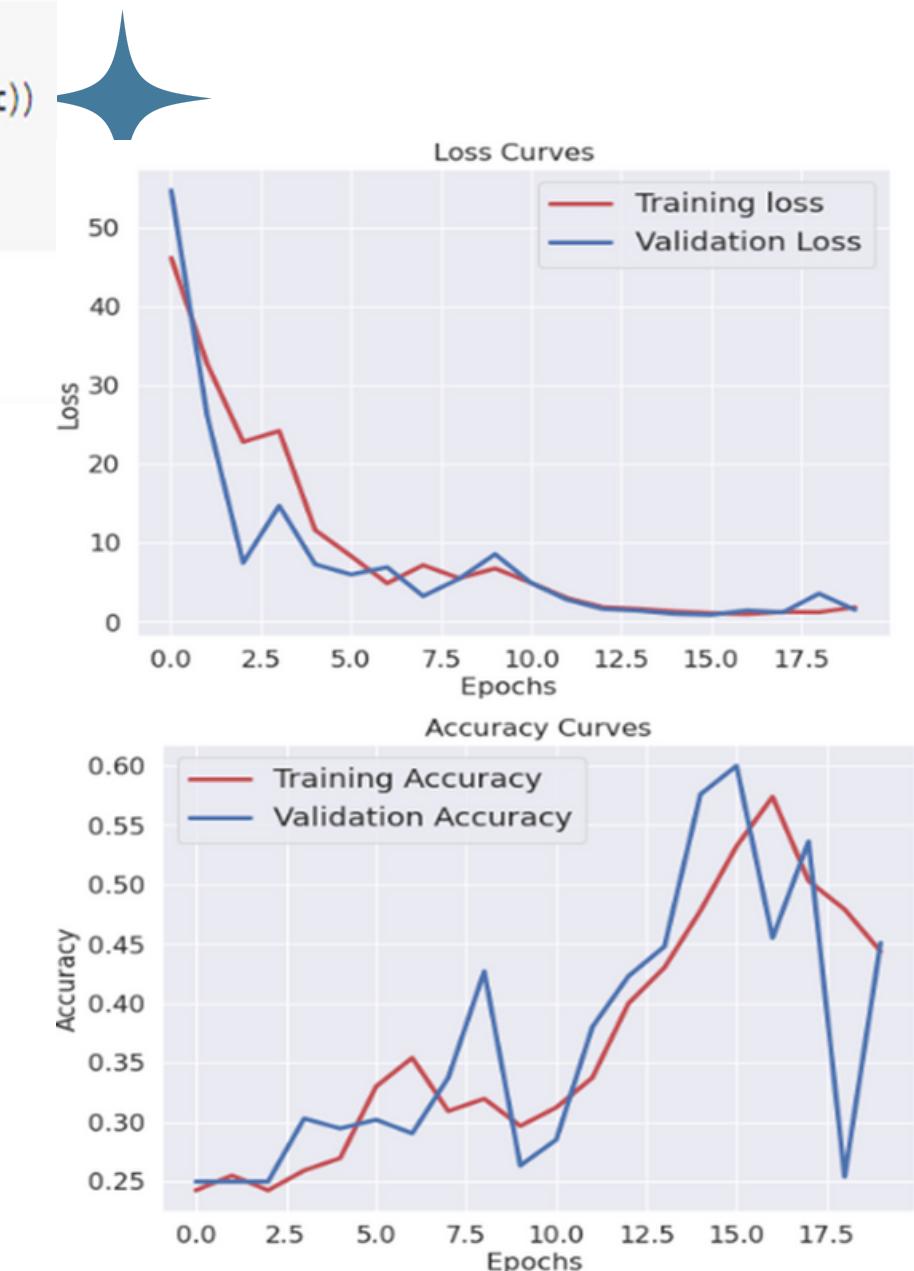
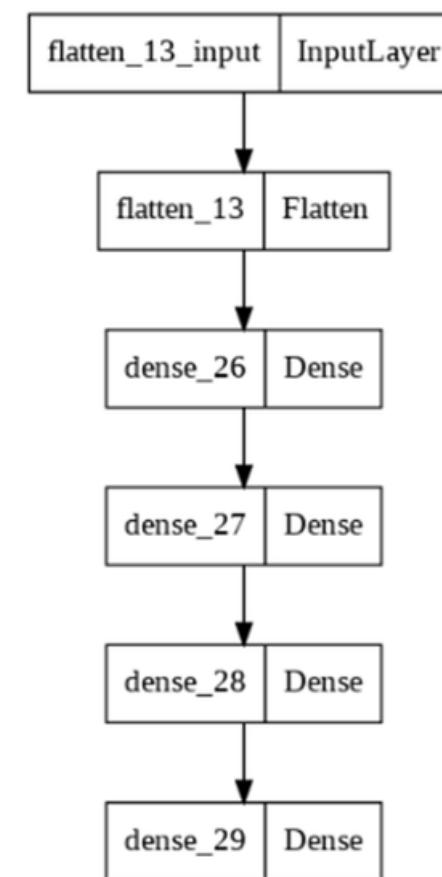
Có nhiều sự nhầm lẫn xuất hiện nhiều ở cặp eraser và pencil, cũng xảy ra ở pencil vs marker, v.v có thể hiểu là do nó khá giống nhau

MLP Model

Dự đoán của mô hình

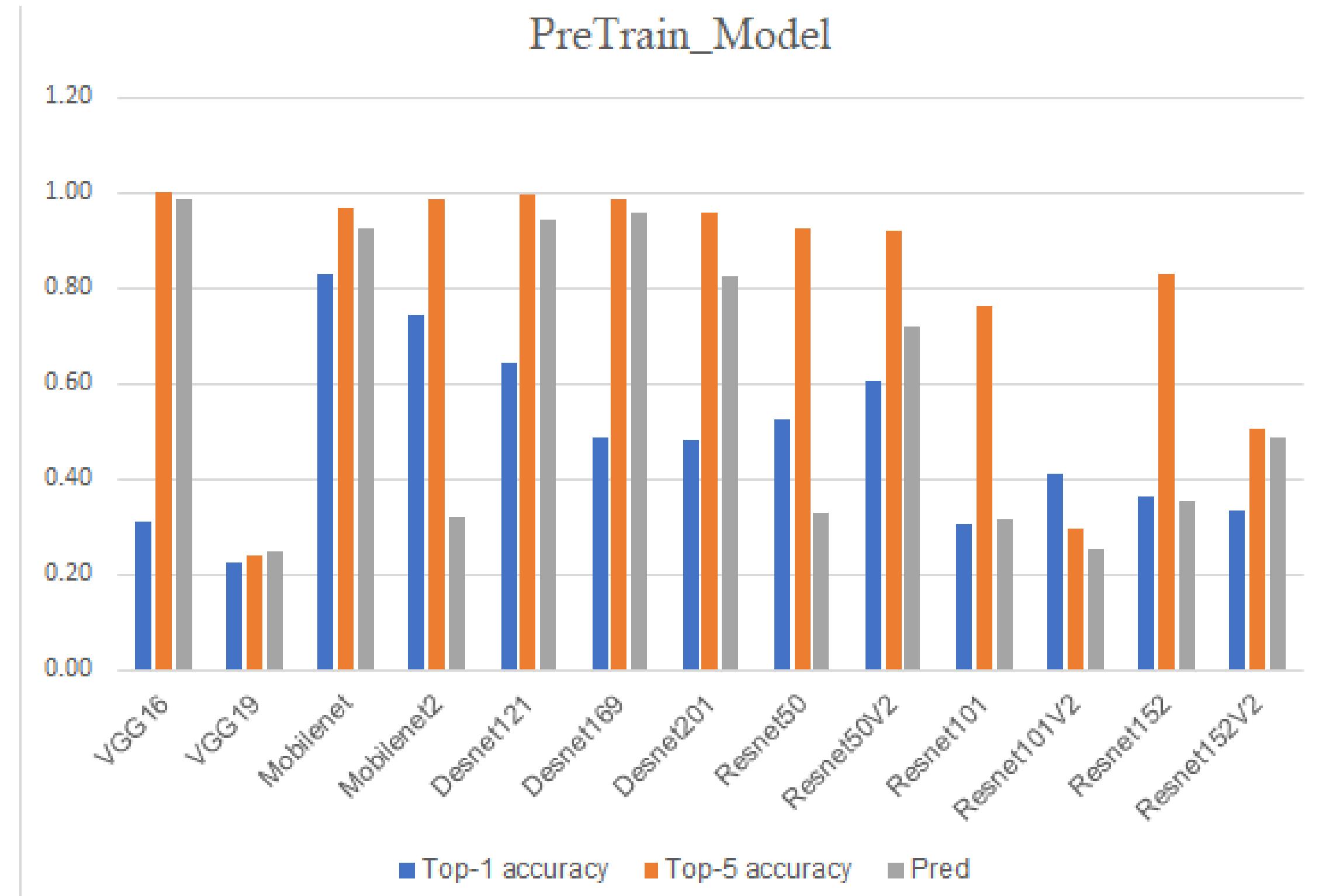
```
[ ] [test_loss, test_acc] = model.evaluate(test_images, test_labels)
print("Evaluation result on Test Data : Loss = {}, accuracy = {}".format(test_loss, test_acc))
```

```
8/8 [=====] - 0s 12ms/step - loss: 1.8170 - accuracy: 0.4000
Evaluation result on Test Data : Loss = 1.8170130252838135, accuracy = 0.4000000059604645
```



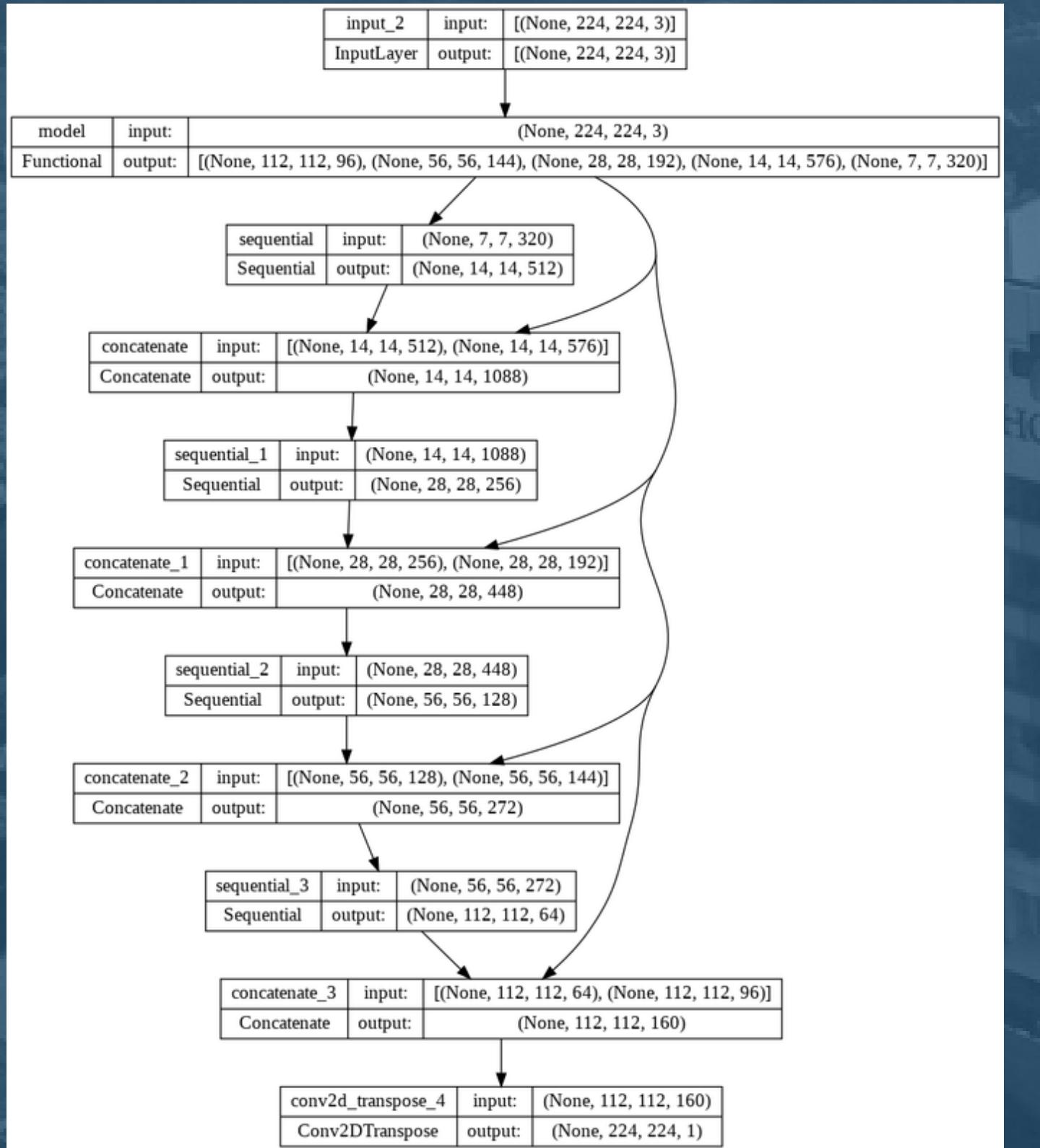
	image_size = 224,224	channels = 3										
	batch_size	epochs	optimizer	loss	metrics	validation_split	accuracy	val_accuracy	loss	val_loss	predictions	
CNN	128	20	adam	categorical_crossentropy	accuracy	20%	0.51	0.52	1.17	1.09	0.40	
MLP	128	30	adam	sparse_categorical_crossentropy	accuracy	20%	1.00	0.72	0.01	1.11	0.75	

Pretrain Model



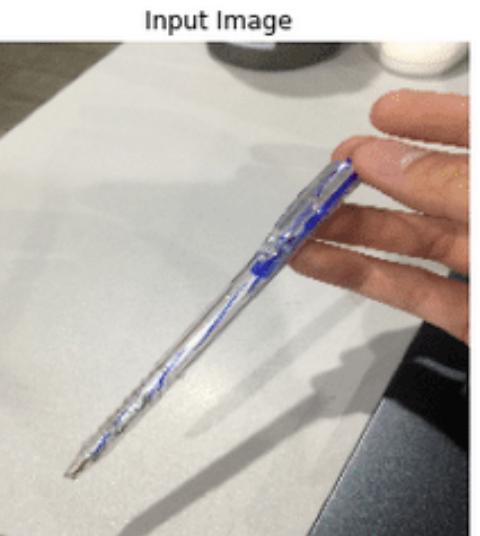


Segmentation với U-net



Dự đoán của mô hình

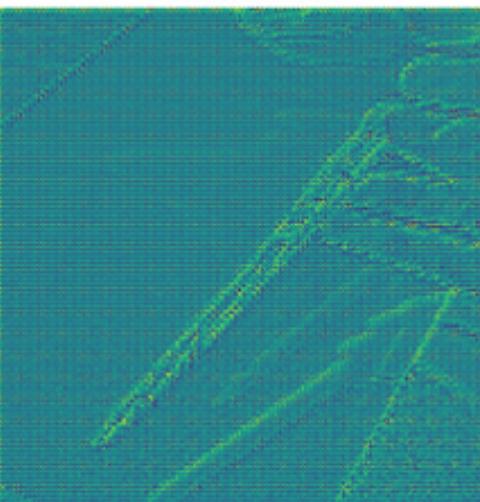
1/1 [=====] - 3s 3s/step



True Mask



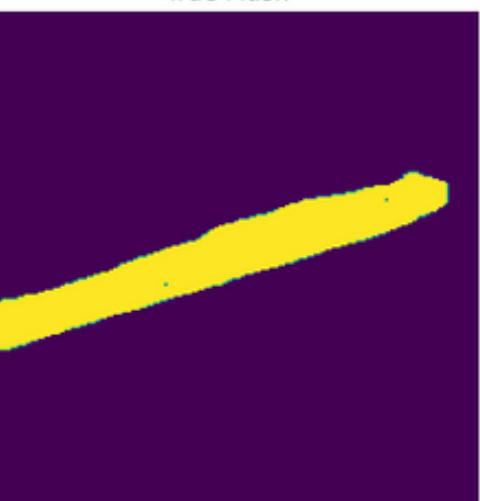
Predicted Mask



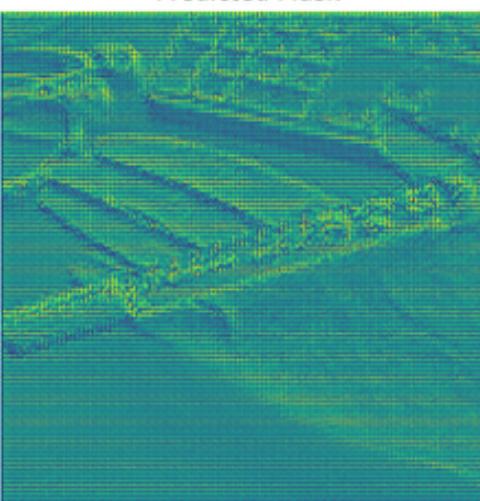
1/1 [=====] - 9s 9s/step



True Mask



Predicted Mask

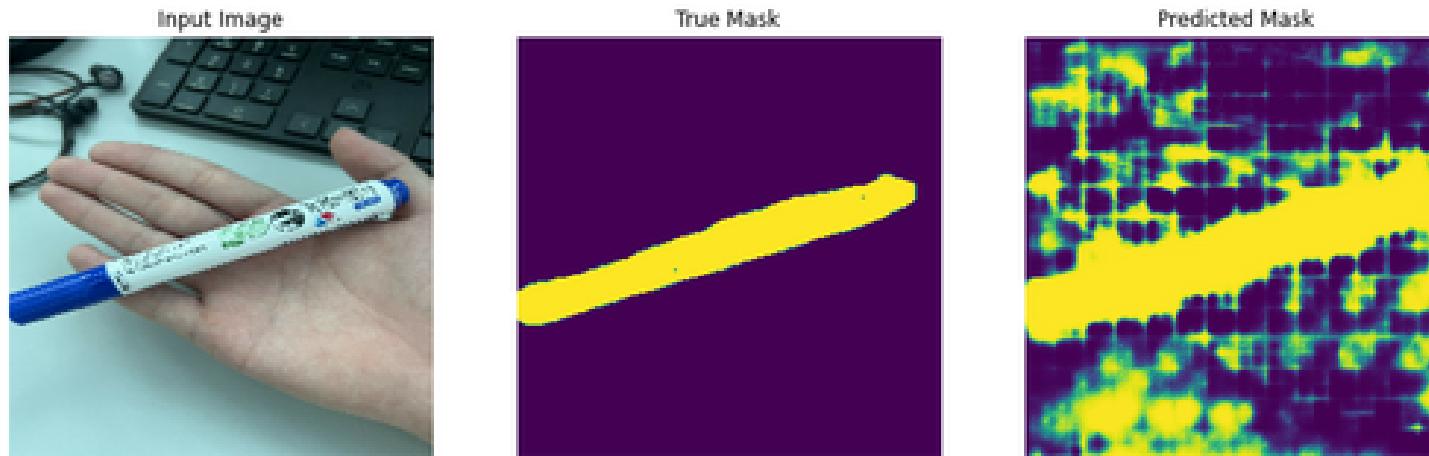




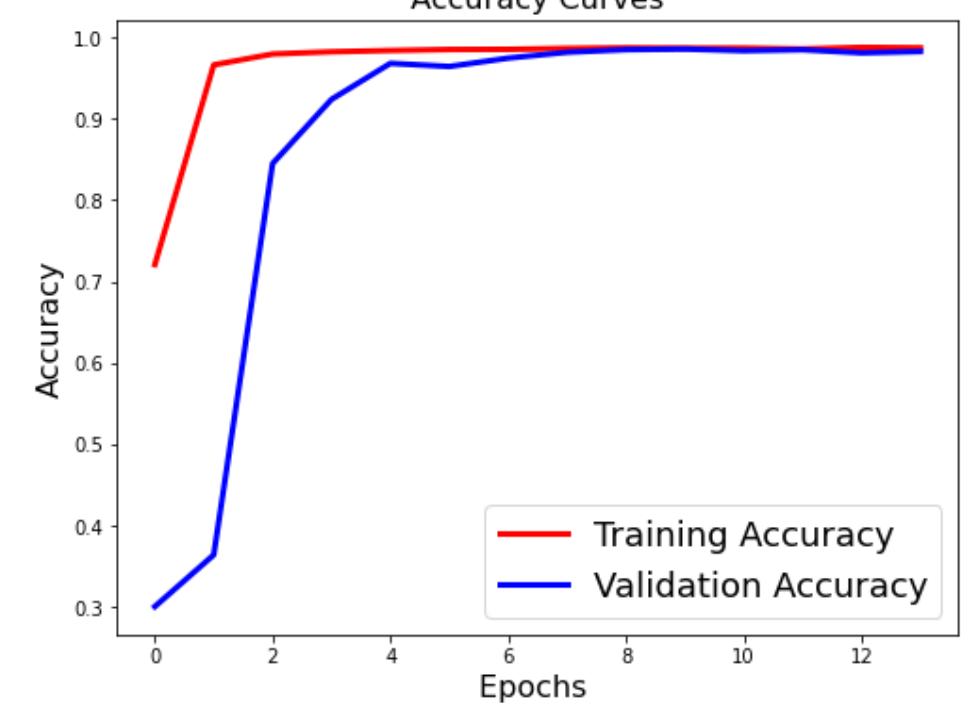
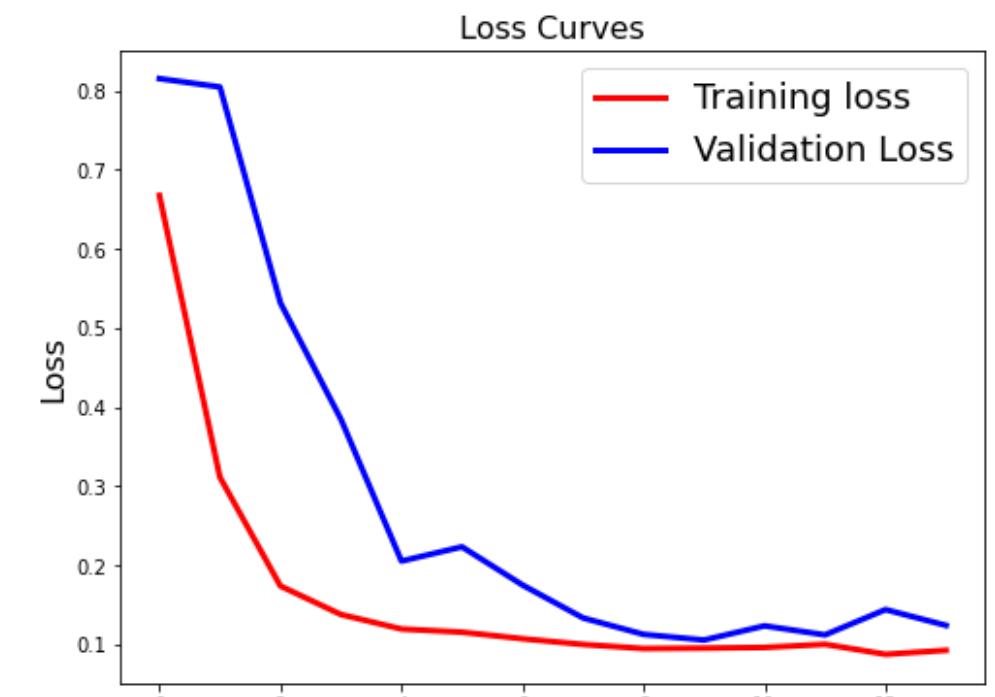
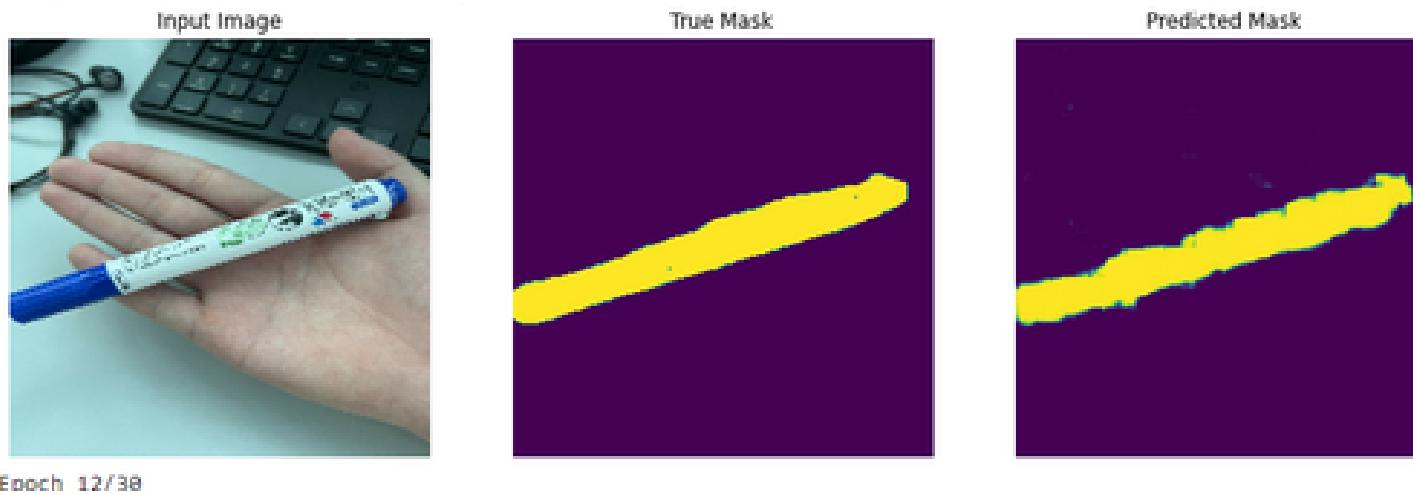
Train dữ liệu

Quá trình train của mô hình

```
Epoch 1/30  
14/14 [=====] - 32s 2s/step - loss: 0.6672 - dice_coef: 0.3328 - accuracy: 0.7207 - val_loss: 0.8149 - val_dice_coef: 0.1838 - val_accuracy: 0.3006  
Epoch 2/30  
14/14 [=====] - 4s 308ms/step - loss: 0.3111 - dice_coef: 0.6796 - accuracy: 0.9661 - val_loss: 0.8843 - val_dice_coef: 0.1958 - val_accuracy: 0.3641  
1/1 [=====] - 0s 28ms/step
```



```
Epoch 9/30  
14/14 [=====] - 4s 297ms/step - loss: 0.0947 - dice_coef: 0.9831 - accuracy: 0.9878 - val_loss: 0.1126 - val_dice_coef: 0.8866 - val_accuracy: 0.9848  
Epoch 10/30  
14/14 [=====] - 4s 300ms/step - loss: 0.0953 - dice_coef: 0.9863 - accuracy: 0.9867 - val_loss: 0.1055 - val_dice_coef: 0.8940 - val_accuracy: 0.9855  
Epoch 11/30  
14/14 [=====] - 4s 289ms/step - loss: 0.0961 - dice_coef: 0.8984 - accuracy: 0.9867 - val_loss: 0.1234 - val_dice_coef: 0.8768 - val_accuracy: 0.9835  
1/1 [=====] - 0s 19ms/step
```





**Thank You
For Your Time!**