

Họ tên: Trần Văn Anh

MSV: B20DCCN075

1. Load dữ liệu về máy

```
import cv2
import os
import numpy as np
import pandas as pd

# Đường dẫn đến thư mục chứa hình ảnh
root_folder = "assets"
# Tạo danh sách hình ảnh và nhãn sử dụng list comprehension
data = []
labels = []
for label in ["That tinh", "Dang yeu", "Tuong tu"]:
    label_folder = os.path.join(root_folder, label)

    if os.path.isdir(label_folder): # Kiểm tra thư mục tồn tại
        images = [img for img in os.listdir(label_folder) if img.lower().endswith((".jpg", ".png"))]

        for filename in images:
            image_path = os.path.join(label_folder, filename) # Đường dẫn đến ảnh

            # Đọc hình ảnh bằng OpenCV
            img = cv2.imread(image_path, cv2.IMREAD_COLOR)
            if(type(img) == type(None)):
                pass
            else:
                # Đảm bảo rằng hình ảnh có kích thước ít nhất 64x64
                img = cv2.resize(img, (64, 64))
                # Đảm bảo rằng hình ảnh có 3 kênh màu (RGB)
                img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
                # Chuyển đổi ảnh thành mảng NumPy
                img_array = np.array(img)

                data.append(img_array)
                labels.append(label)

# Chuyển danh sách thành mảng NumPy
data = np.array(data)
labels = np.array(labels)

# Tạo DataFrame pandas với hình ảnh và nhãn
df = pd.DataFrame({'Image': data.tolist(), 'Label': labels})

# Lưu DataFrame vào tệp CSV
df.to_csv('Test.csv', index=False)
print(df.shape)
```

(768, 2)

```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import ast
# Đọc tệp CSV
df = pd.read_csv('Test.csv')
# Lặp qua cột 'Image' và 'Label' của 5 dòng đầu tiên
for image_data, label in zip(df['Image'].head(5), df['Label'].head(5)):
    # Chuyển chuỗi số thành mảng NumPy
    image_data = np.array(ast.literal_eval(image_data))
    # Hiển thị hình ảnh
    plt.imshow(image_data)
    plt.title(label)
    plt.show()

```

```

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
import numpy as np
import pandas as pd

# Đọc tệp CSV
df = pd.read_csv('Test.csv')

# Chuyển đổi dữ liệu hình ảnh từ chuỗi thành mảng NumPy
data = df['Image'].apply(lambda x: np.array(eval(x))).values
labels = df['Label'].values

# Kết hợp tất cả các mảng hình ảnh thành một tensor
X_data = np.stack(data, axis=0)

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X_data, labels, test_size=0.2, random_state=32)

# Chuẩn hóa giá trị pixel trong khoảng [0, 1]
X_train = X_train / 255.0
X_test = X_test / 255.0

# Mã hóa one hot cho nhãn
label_binarizer = LabelBinarizer()
y_train_one_hot = label_binarizer.fit_transform(y_train)
y_test_one_hot = label_binarizer.transform(y_test)

```

1. Xây dựng 3 mô hình dự đoán với kiến trúc (thay đổi số layer và neuron) và dropout khác nhau cho phân loại ảnh trên
 - Mô hình 1

```

# Model 1

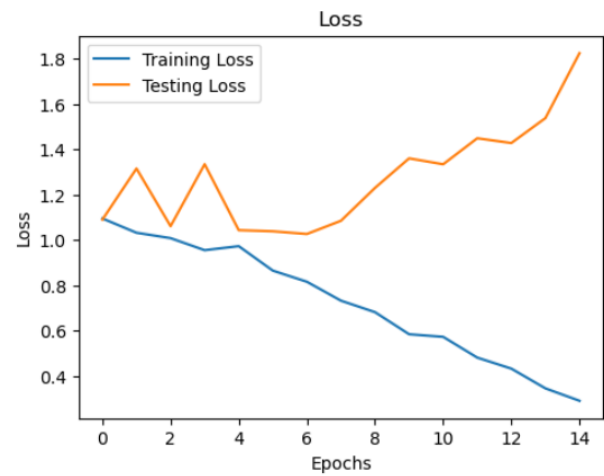
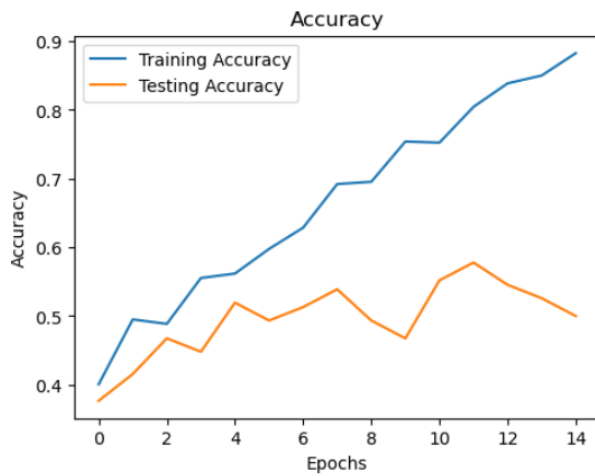
# Xây dựng mô hình CNN
model = keras.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dropout(0.5),
    layers.Dense(128, activation='relu'),
    layers.Dense(3, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
# Huấn Luyện mô hình
history = model.fit(X_train, y_train_one_hot, epochs=15, batch_size=10, validation_data=(X_test, y_test_one_hot))

```

```

# Model 1
import matplotlib.pyplot as plt
# Lấy giá trị accuracy và loss từ biến "history"
train_accuracy = history.history['accuracy']
test_accuracy = history.history['val_accuracy']
train_loss = history.history['loss']
test_loss = history.history['val_loss']
# Vẽ biểu đồ
plt.figure(figsize=(12, 4))
# Biểu đồ Accuracy
plt.subplot(1, 2, 1)
plt.plot(train_accuracy, label='Training Accuracy')
plt.plot(test_accuracy, label='Testing Accuracy')
plt.legend()
plt.title('Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
# Biểu đồ Loss
plt.subplot(1, 2, 2)
plt.plot(train_loss, label='Training Loss')
plt.plot(test_loss, label='Testing Loss')
plt.legend()
plt.title('Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.show()
# Đánh giá mô hình trên tập kiểm tra
test_loss, test_accuracy = model.evaluate(X_test, y_test_one_hot)
print(f"Test Accuracy: {test_accuracy * 100:.2f}%")

```



5/5 [=====] - 1s 91ms/step - loss: 1.8234 - accuracy: 0.5000
 Test Accuracy: 50.00%

- Mô hình 2

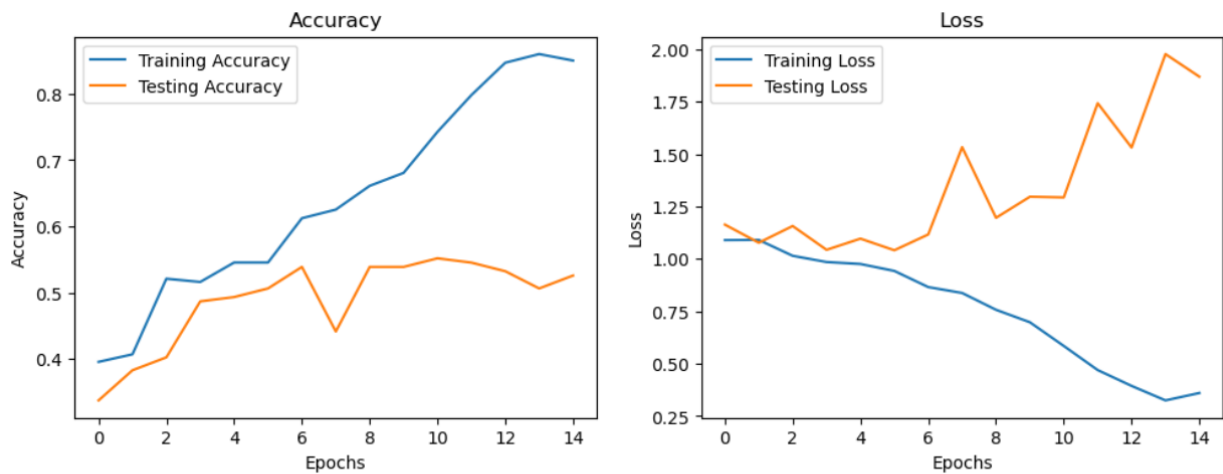
```
# Model 2

# Xây dựng mô hình CNN
model = keras.Sequential([
    layers.Conv2D(64, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(256, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dropout(0.3), # Giảm dropout để giảm khả năng overfitting
    layers.Dense(256, activation='relu'),
    layers.Dense(3, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
# Huấn luyện mô hình
history = model.fit(X_train, y_train_one_hot, epochs=15, batch_size=10, validation_data=(X_test, y_test_one_hot))
```

```

# Model 2
import matplotlib.pyplot as plt
# Lấy giá trị accuracy và loss từ biến "history"
train_accuracy = history.history['accuracy']
test_accuracy = history.history['val_accuracy']
train_loss = history.history['loss']
test_loss = history.history['val_loss']
# Vẽ biểu đồ
fig, axes = plt.subplots(1, 2, figsize=(12, 4))
# Biểu đồ Accuracy
axes[0].plot(train_accuracy, label='Training Accuracy')
axes[0].plot(test_accuracy, label='Testing Accuracy')
axes[0].legend()
axes[0].set_title('Accuracy')
axes[0].set_xlabel('Epochs')
axes[0].set_ylabel('Accuracy')
# Biểu đồ Loss
axes[1].plot(train_loss, label='Training Loss')
axes[1].plot(test_loss, label='Testing Loss')
axes[1].legend()
axes[1].set_title('Loss')
axes[1].set_xlabel('Epochs')
axes[1].set_ylabel('Loss')
plt.show()
# Đánh giá mô hình trên tập kiểm tra
test_loss, test_accuracy = model.evaluate(X_test, y_test_one_hot)
print(f"Test Accuracy: {test_accuracy * 100:.2f}%")

```



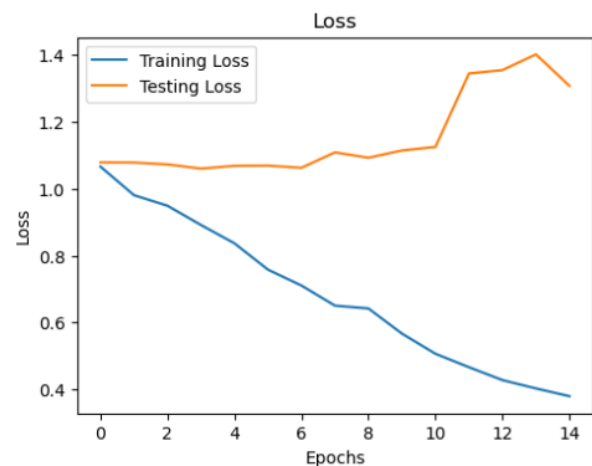
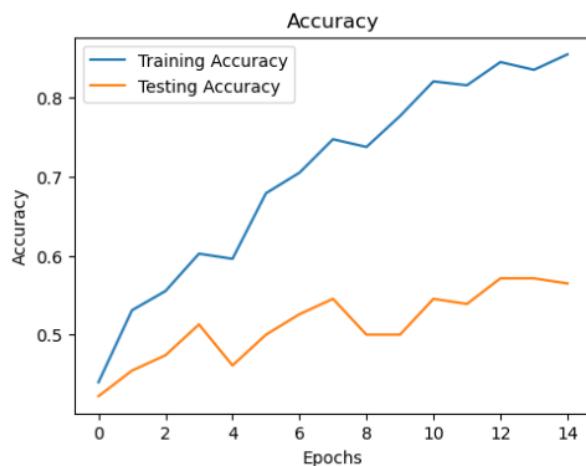
5/5 [=====] - 0s 38ms/step - loss: 1.8688 - accuracy: 0.5260
Test Accuracy: 52.60%

- Mô hình 3

```
# Model 3

# Xây dựng mô hình CNN
model = keras.Sequential([
    layers.Conv2D(8, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(16, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dropout(0.4),
    layers.Dense(16, activation='relu'),
    layers.Dense(3, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
# Huấn luyện mô hình
history = model.fit(X_train, y_train_one_hot, epochs=15, batch_size=10, validation_data=(X_test, y_test_one_hot))
```

```
: # Model 3
import matplotlib.pyplot as plt
# Lấy giá trị accuracy và loss từ biến "history"
train_accuracy = history.history['accuracy']
test_accuracy = history.history['val_accuracy']
train_loss = history.history['loss']
test_loss = history.history['val_loss']
# Vẽ biểu đồ
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(train_accuracy, label='Training Accuracy')
plt.plot(test_accuracy, label='Testing Accuracy')
plt.legend()
plt.title('Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.subplot(1, 2, 2)
plt.plot(train_loss, label='Training Loss')
plt.plot(test_loss, label='Testing Loss')
plt.legend()
plt.title('Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.show()
# Đánh giá mô hình trên tập kiểm tra
test_loss, test_accuracy = model.evaluate(X_test, y_test_one_hot)
print(f"Test Accuracy: {test_accuracy * 100:.2f}%")
```



5/5 [=====] - 0s 11ms/step - loss: 1.3080 - accuracy: 0.5649
Test Accuracy: 56.49%

2. Đánh giá độ chính xác và mất mát của 3 mô hình

- Về độ chính xác của các model giảm dần theo model 3, model 2, model 1
- Tỷ lệ loss của các model tăng dần theo model 1, mode 2 và model 3
- Cả 3 model đều bị overfitting
- Sự khác biệt về hiệu suất giữa ba mô hình có thể được giải thích bằng nhiều yếu tố, bao gồm cấu trúc mô hình, tỷ lệ dropout, số lượng lớp và neurons, và cách mô hình phản ứng với tập dữ liệu cụ thể.