Le Thi Huyen University of Economics Ho Chi Minh City 31211026505

**SCIENTIFIC ARTICLE**

**PREDICTING GENDER AND EMOTIONS BASED ON FACIAL FEATURES (REAL-TIME USING A CAMERA)**

**Le Thi Huyen1**

*1* University of Economics Ho Chi Minh City

**Lecturers:** PGS.TS Nguyen Truong Thinh

**Subject:** A.I in Business

**Workshop:** 23D1TEC55004001

**Periods:** Thursday (2-6)

## SUMMARY

In the age of intelligence, artificial intelligence is rapidly advancing and revolutionizing various fields. One particular area that has garnered significant attention is the simultaneous prediction of gender and emotion based on facial analysis, especially with the growing influence of social media and communication platforms. This research paper introduces an innovative approach utilizing convolutional neural networks (CNN) to tackle this challenge.

By emulating the intricate neural networks found in humans and animals, CNNs provide a powerful tool for extracting meaningful features from facial images. The proposed method trains and deploys CNN models capable of accurately predicting both gender and emotion. The models were trained using a diverse dataset gathered from various online sources.The experimental results are highly promising, showcasing the effectiveness of the CNN models.

The gender prediction model achieves an impressive accuracy of 99% on the training set and 96% on the test set. For emotion prediction, the model achieves a training set accuracy of 54% and a test set accuracy of 50%. These results demonstrate the potential of CNNs in accurately capturing gender and emotion information from facial expressions.

Notably, the developed models exhibit real-time performance, making them suitable for real-world facial recognition applications. The combination of advanced artificial intelligence techniques and CNNs holds great promise in enhancing our understanding and practical applications of gender and emotion recognition.

Overall, this research sheds light on the immense possibilities offered by artificial intelligence and convolutional neural networks in the realm of gender and emotion recognition, paving the way for future advancements in the field.

**Key word:***Convolutional neutal network (CNN)****,*** *Gender Classifier, Emotion Classifier, Age Classifier, Deep Learning, Neural Network.*

### Receipt date: May 21, 2022; Completion date 4/6/2022; Submission date: 6/6/2022

*For any feedback please contact:Phone: 0812375083, Email: huyenle.31211026505@st.ueh.edu.vn*

# Introduce

Applications of image processing are interesting fields that attract the most attention in the artificial intelligence industry with many highly applicable problems in practice. At the same time, with the strong development of deep learning algorithms, especially convolutional neural networks (CNN), have given outstanding results in testing problems. For example, in 2015, Kaiming He [2] proposed the ResNet network architecture and achieved a very good error rate of only 3.57%. In 2012, Alex and his research colleagues [1] proposed a model using CNN network and won the ImageNet competition with an error rate of 15%. A contest with a very large scale on the problem of object recognition and detection in images.

In humans, the face can express emotional states, gender. In psychology, recognizing emotions and gender from faces plays an important role in many researches and applications such as: Emotion recognition, Adaptive psychology research, Research on social perception Society, Gender Analysis. Therefore, the determination of emotions and gender based on faces is an important and meaningful problem and the applicability to reality is extremely large.

# Recent research

Machine learning has been proven to be very good at classification problems, and one of the subsets of machine learning is deep learning. The problem of gender, emotion and gender recognition is one of the typical problems using the CNN algorithm of research intelligence that has been implemented for a long time. However, all previous studiesonly stop at studying each feature alone, not combining the two characteristics above. And in this paper we will present the methods of building each individual model and the model results. First, we will go through each problem in turn to have the most general overview of the reseach related to this paper.

***2.1 Emotion prediction problem:***

The problem of emotion recognition and prediction has been used in many companies to evaluate service quality when analyzing the emotional states of customers while interacting with their products. At ParallelDots, they have applied psychological science, human emotion representation and artificial intelligence to identify emotions

on the face. Or at Unilever company, which has combined facial emotion detection technology in interviews, with this technology, recruiters will know the confidence level of candidates and make a decision whether the candidate Is this employee suitable for a customer-facing job…

***2.2 Gender prediction problem:***

In addition to the strong development of age and emotion prediction problems, the problem of gender recognition was proposed in 1990 and researchers built gender prediction models. simple. And recently, there is a very good study on artificial intelligence that identifies male and female gender through smiles with an accuracy of up to 86%. This method considered the facial movement when smiling by tracking the movements of 49 points on the face and landmarks near locations such as eyes and mouth [3].

However, after this study was completed, questions were raised about gender identity and transgender people. At the same time, the research team believes that facial expressions can help identify gender such as expressions of anger, surprise, disgust, etc.

# Algorithm Proposal

In this section, a method of using convolutional neural networks (CNN) will be presented to solve the problem of predicting gender and emotions. To make it easier for readers to understand, I will present it in 3 parts including: a brief introduction to the algorithm used, the implementation method and finally the real-time identification results. In the following, we will introduce the convolutional neural network (CNN).

## Convolutional Neural Networks CNN

* + 1. ***Introduce about CNN:***

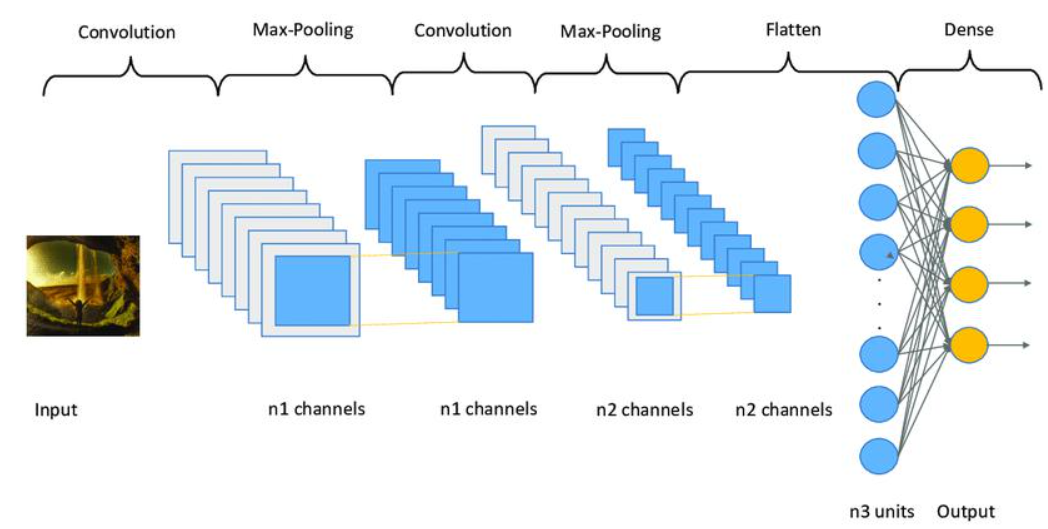
Convolutional Neural Networks (CNNs) are a powerful network architecture specifically designed for image recognition and classification tasks. Among the various applications of CNNs, one prominent area is the recognition of human faces. With their ability to learn and extract meaningful features from images, CNNs have been widely employed in the field of face recognition. They excel at capturing intricate facial details and patterns, allowing for accurate identification of individuals based on their facial characteristics. Through the use of convolutional layers, pooling layers, and fully connected layers, CNNs enable us to leverage the power of deep learning to tackle the challenges of facial recognition with impressive accuracy and efficiency.

Convolutional neural network (CNN) is a method used to classify images by processing and labeling images based on their content. When fed to an image, CNN views it as an array of pixels, in which information about the image's height, width, and depth is determined.

A picture containing rectangle, screenshot, diagram, colorfulness

Description automatically generated

The CNN model is used for training and evaluation, where each input image is passed through a series of convolutional layers with filters (kernels). Then, through fully connected layers and using Softmax function, the model classifies objects with probability values ​​between 0 and 1. Through this process, the model is able to understand and receive identify important features in the image and use them to determine the emotion or gender of the face.



***Image 2.*** CNN network structure

***3.2.2 Convolution Layer:***

Convolution is an important class in convolutional neural networks (CNNs) used to extract features from input images. The convolution layer keeps the relationship between pixels in the image by applying small filters on the input data.

To understand how the convolution layer works, consider a 5x5 image matrix with pixel values ​​of 0 and 1. At the same time, we also have a 3x3 filter matrix. By multiplying the image matrix and the filter matrix together, we create a Feature Map, where each value in the Feature Map represents a feature found in the original image.

Through convolution, CNN learns low-level features like contours and angles, and higher-level features like faces and objects. The convolution layer reduces the size of the input data and creates abstract representations of the images, making the model training more efficient and helping to improve the recognition and classification capabilities.

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

***Image 3.*** Multiply Image Matrix và Filter Matrix

* + 1. ***Stride:***

The jump (Stride) is the distance between consecutive kernel positions on the input matrix. When stride is 1, kernel moves 1 pixel at a time, when stride is 2, kernel moves 2 pixels at a time, and so on for other stride values. On each move, the kernel applies the calculation to the element of the input matrix located in the region corresponding to the kernel.

A picture containing text, screenshot, diagram, font

Description automatically generated

1. ***3.2.4 Padding and Relu:***
2. There are two methods to help us handle when the kernel is not compatible with the input image: inserting zeros into the four edges of the image or removing the parts that do not match the kernel.
3. The ReLU (Rectified Linear Unit) function is a widely used nonlinear function in the construction of a convolutional neural network (CNN). It has an output formula of f(x) = max(0, x), where x is the input. ReLU is a non-negative function and retains a constant value for non-negative values, while converting negative values ​​to 0.

ReLU is favored in CNNs because it solves the nonlinearity of data in ConvNet. The data is usually non-negative linear values, and ReLU helps to keep positive values ​​unchanged while bringing negative

values ​​to zero. Although tanh and sigmoid functions can also be used as an alternative to ReLU, ReLU is still favored because of its better performance in CNN model building.

A picture containing text, rectangle

Description automatically generated

***3.2.5 Pooling Layer:***

In image processing using a convolutional neural network (CNN), an important stage is that the pooling layer is applied after the convolutional layer. The goal of the pooling layer is to reduce the complexity of the output and reduce the number of neurons that need to be computed. The pooling layer performs spatial subsampling, thereby reducing the size of each feature map while preserving the most important information.

There are different types of pooling like Max-Pooling, Average-Pooling, Sum-Pooling, however, in most cases, Max-Pooling is the most common pooling method used. In Max-Pooling, the maximum value in the input region of a feature matrix is ​​chosen to represent that region. This helps to create a scaled-down version of the feature with important information still retained.

In summary, the pooling layer in CNN is used to reduce the output feature size, simplify the model, and retain important information. The most common pooling method is Max-Pooling, where the largest value in each input region is chosen to represent that region.

A picture containing text, screenshot, number

Description automatically generated

Thus, we can see that through the Max Pooling class, the number of neurons is significantly reduced compared to the original. With CNN, there are many Feature Map, so for each Feature Map we need each Max Pooling is different. Max Pooling has the effect of finding the most important features among all the existing features to make the training model the most efficient.

## Building CNN models

### Model building ideas

In the model predicting gender, emotion, I will train 2 speparate models (CNN) as fllows:

-Model 1: That person is Male or Female

-Model2: Identify 5 emotion: 'Neutral', 'Happy', 'Sad', 'Angry', 'Suprised'

After determining the research direction for the topic, the next step is to build a convolutional neural network model CNN using Python programming language and conduct training. When training is complete, we save the model to .h5 format files and then use this model for real-time recognition. The image data used in the study has been collected from Kaggle and Google sources, and then preprocessed according to the requirements and purposes of the model training process.

### Methods of implementation

To implement the model, you need to think ahead of time and the implementation process follows the sequence of steps as shown in the following figure:

A picture containing text, screenshot, font, line

Description automatically generated

***Image 7.*** Model execution process

### Data cllection

- Regarding the gender data set, I collected the data set online with 2 genders, Male and Female, and re-filtered it as desired. Each gender has 2500 photos and a total of 5000 photos for 2 genders.

- About the dataset for emotions, I collected the data set through kaggle and google images with 5 emotions: Neutral, Sad, Happy, Angry, Surprised and filtered as desired. Each emotion has 5000 photos and a total of 25000 photos for 5 emotions.

### Data preprocessing

- Synchronize data size by color image with size 150x150.

- Normalize input data in the range (0,1) instead of 255 as the original.

### Model building

- After data collection and data preprocessing, I proceed to build a gender prediction model.

A picture containing text, font, receipt, diagram

Description automatically generatedA picture containing text, screenshot, font, receipt

Description automatically generated

***Image 8.*** CNN model predicts emotion

A picture containing text, screenshot, font, parallel

Description automatically generatedA picture containing text, font, screenshot, receipt

Description automatically generated

***Image 9.*** CNN model predicts gender

### The result of the model

* ***Model self assessment***

After completing the process of building and training the model, the obtained results show high accuracy in predicting gender and emotion from the face. For the gender prediction model, the model achieved accuracy up to 93% on the training dataset and 91% on the validation dataset. This shows that the model is capable of recognizing gender from the face with a reliable level of accuracy.

For the emotion prediction model, although the accuracy is not as high, it still has a significant degree of accuracy. The model achieved an accuracy of 53% on the training dataset and 50% on the validation dataset. Although this level of accuracy may need to be improved, it shows that the model is still capable of recognizing emotions from faces with a degree of reliability.

A picture containing text, plot, screenshot, line

Description automatically generated

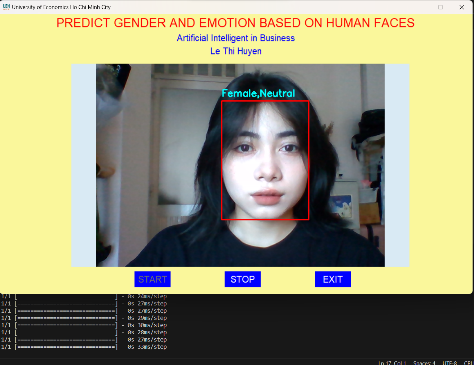
A picture containing text, plot, screenshot, line

Description automatically generated

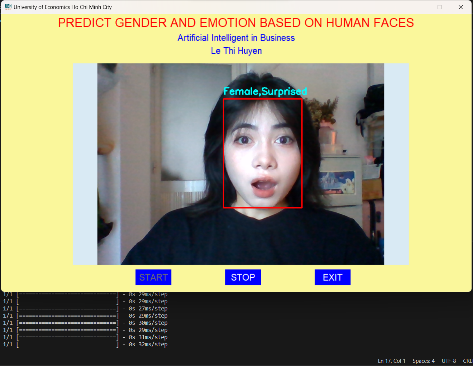
In summary, the model has achieved high accuracy in predicting gender and significant accuracy in predicting emotions from faces. These results demonstrate the effectiveness of the model in applying artificial intelligence to analyze and recognize psychological information from faces.

### Realtime experimental results

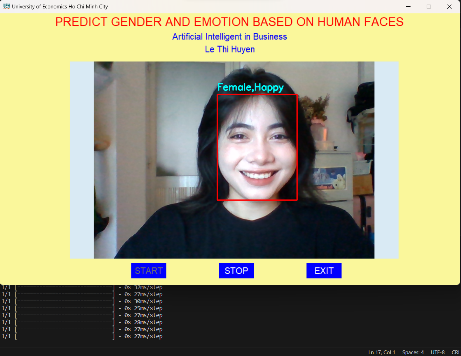
After the model has been trained and checked for accuracy, the next step is to perform a quality assessment of the model in real time. To do this, we store the h5 files of the model and integrate them into the realtime source code. Through the process of running in real time, we can evaluate the performance of the model and get the experimental results



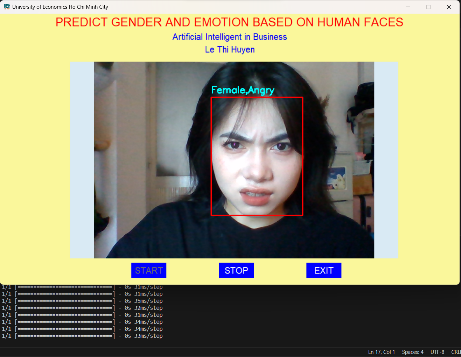
***Image 11.*** Realtime running results 1



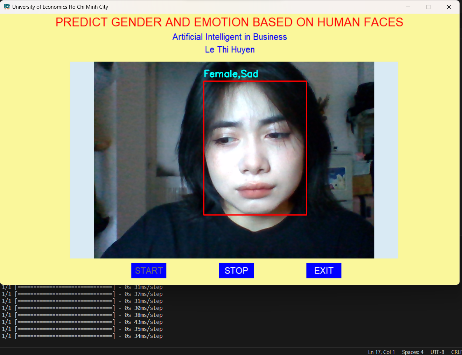
***Image 12.*** Realtime running results 2



***Image 13.*** Realtime running results 3



***Image 14.*** Realtime running results 4



***Image 15 .*** Realtime running results 5

# Conclusion and development direction

In this study, we proposed a deep learning model for gender and emotion recognition based on face images. The model is built using CNN convolutional neural network layers and has been tested multiple times to optimize performance. In particular, the model is designed to work in real time, capable of recognizing faces as soon as an image is provided.

In the future, we are continuing to develop the project to bring it to market and commercialize. Part of the next work is to use pre-trained network layers to improve the accuracy and predictive quality of the model. This will help improve real-time facial recognition and better respond to real-world applications.

**Acknowledgements**

Lastly, I would like to sincerely thank Prof. Dr. Nguyen Truong Thinh for imparting fundamental knowledge throughout the learning process, which greatly assisted me in completing this project. Additionally, I would like to express my gratitude to assistant lecturer Minh Trieu for providing timely support whenever I had questions during the project.

I have endeavored to apply the theoretical knowledge learned in class and incorporate feedback from my instructor to complete this report. However, due to limited practical experience, there may still be some shortcomings in my report. I kindly request the evaluation and constructive feedback from everyone.

# References

[1]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[2]. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and

A. Rabinovich, “Going deeper with convolutions”, *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 1–9, 2015.

[3]. KTHN, “*AI có thể nhận diện giới tính nam nữ qua nụ cười*”, [https://kthn.edu.vn/ai-co-the-](https://kthn.edu.vn/ai-co-the-nhan-dien-gioi-tinh-nam-nu-qua-nu-cuoi/) [nhan-dien-gioi-tinh-nam-nu-qua-nu-cuoi/,](https://kthn.edu.vn/ai-co-the-nhan-dien-gioi-tinh-nam-nu-qua-nu-cuoi/) truy cập ngày 20/6/2022.

[4]. Chung Pham Van, “*[Deep Learning] Tìm hiểu về mạng tích chập (CNN)*”, [https://viblo.asia/p/deep-learning-tim-hieu-ve-](https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2) [mang-tich-chap-cnn-maGK73bOKj2,](https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2) truy cập ngày 10/6/2022.

[5]. Y. Kim, “Convolutional neural networks for sentence classification,” arXiv preprint arXiv, 1408.5882, 2014.

**Appendix**

***Modeling and training the emotional dataset model***

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization

from tensorflow.keras.optimizers import RMSprop

from tensorflow.keras.callbacks import EarlyStopping

# Tạo bộ tăng cường dữ liệu (data augmentation)

train\_datagen = ImageDataGenerator(rescale=1.0/255,

rotation\_range=20,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

fill\_mode='nearest')

# Áp dụng data augmentation cho tập huấn luyện

train\_generator = train\_datagen.flow\_from\_directory(directory="/kaggle/input/emotion/train\_class",

target\_size=(150, 150),

batch\_size=32,

class\_mode='categorical')

# Tạo bộ tăng cường dữ liệu cho tập validation (không thay đổi dữ liệu)

validation\_datagen = ImageDataGenerator(rescale=1.0/255)

# Áp dụng data augmentation cho tập validation

validation\_generator = validation\_datagen.flow\_from\_directory(directory="/kaggle/input/emotion/val\_class",

target\_size=(150, 150),

batch\_size=32,

class\_mode='categorical')

# Khởi tạo mô hình CNN

model = Sequential()

# Các lớp Convolutional

model.add(Conv2D(16, (3, 3), activation='relu', input\_shape=(150, 150, 3)))

model.add(MaxPooling2D(2, 2))

model.add(Conv2D(32, (3, 3), activation='relu'))

model.add(MaxPooling2D(2, 2))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(2, 2))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D(2, 2))

model.add(Conv2D(256, (3, 3), activation='relu'))

model.add(MaxPooling2D(2, 2))

# Flatten

model.add(Flatten())

# Fully-connected layers

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.5))

# Output layer

model.add(Dense(5, activation='softmax'))

# Compile mô hình

model.compile(optimizer=RMSprop(learning\_rate=0.001),

loss='categorical\_crossentropy',

metrics=['accuracy'])

# Sử dụng EarlyStopping để dừng quá trình huấn luyện sớm nếu không có cải tiến đáng kể

early\_stopping = EarlyStopping(patience=3, monitor='val\_loss', restore\_best\_weights=True)

# Huấn luyện mô hình

history = model.fit(train\_generator,

steps\_per\_epoch=train\_generator.n // train\_generator.batch\_size,

epochs=20,

validation\_data=validation\_generator,

validation\_steps=validation\_generator.n // validation\_generator.batch\_size,

callbacks=[early\_stopping])

# Đánh giá độ chính xác của mô hình

score = model.evaluate(train\_generator,verbose=0)

print('Sai số kiểm tra là: ',score[0])

print('Độ chính xác kiểm tra là: ',score[1])

import matplotlib.pyplot as plt

# Draw plot

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epochs')

plt.legend(['train','Validation'])

plt.show()

plt.plot(history.history['loss'],'r',label='training loss')

plt.plot(history.history['val\_loss'],label='validation loss')

plt.xlabel('# epochs')

plt.ylabel('loss')

plt.legend()

plt.show()

from tensorflow.keras.models import load\_model

model.save('Emotion1.h5')

model\_CNN = load\_model('Emotion1.h5')

from tensorflow.keras.utils import load\_img, img\_to\_array

import matplotlib.pyplot as plt

import numpy as np

filename = "/kaggle/input/emotion-dataset/3. Sad/S (10).jpg"

predict = ['Neutral','Happy','Sad','Angry','Suprised']

predict = np.array(predict)

img = load\_img(filename,target\_size=(150,150))

plt.imshow(img)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

img = img/255

result = np.argmax(model\_CNN.predict(img),axis=-1)

predict[result]

***Modeling and training the gender dataset model***

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization

from tensorflow.keras.optimizers import RMSprop

from tensorflow.keras.callbacks import EarlyStopping

# Tạo bộ tăng cường dữ liệu (data augmentation)

train\_datagen = ImageDataGenerator(rescale=1.0/255,

rotation\_range=20,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

fill\_mode='nearest')

# Áp dụng data augmentation cho tập huấn luyện

train\_generator = train\_datagen.flow\_from\_directory(directory="/kaggle/input/gender",

target\_size=(150, 150),

batch\_size=32,

class\_mode='categorical')

# Tạo bộ tăng cường dữ liệu cho tập validation (không thay đổi dữ liệu)

validation\_datagen = ImageDataGenerator(rescale=1.0/255)

# Áp dụng data augmentation cho tập validation

validation\_generator = validation\_datagen.flow\_from\_directory(directory="/kaggle/input/gender",

target\_size=(150, 150),

batch\_size=32,

class\_mode='categorical')

# Khởi tạo mô hình CNN

model = Sequential()

# Các lớp Convolutional

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(150, 150, 3))) # out = (kichthuocanh - (kernel - 1)) / s

model.add(MaxPooling2D(2, 2))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(2, 2))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D(2, 2))

model.add(Conv2D(256, (3, 3), activation='relu'))

model.add(MaxPooling2D(2, 2))

# Flatten

model.add(Flatten())

# Fully-connected layers

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.25))

# Output layer

model.add(Dense(2, activation='softmax'))

# Compile mô hình

model.compile(optimizer=RMSprop(learning\_rate=0.001),

loss='categorical\_crossentropy',

metrics=['accuracy'])

# Sử dụng EarlyStopping để dừng quá trình huấn luyện sớm nếu không có cải tiến đáng kể

early\_stopping = EarlyStopping(patience=3, monitor='val\_loss', restore\_best\_weights=True)

# Huấn luyện mô hình

history = model.fit(train\_generator,

steps\_per\_epoch=train\_generator.n // train\_generator.batch\_size,

epochs=20,

validation\_data=validation\_generator,

validation\_steps=validation\_generator.n // validation\_generator.batch\_size,

callbacks=[early\_stopping])

# Đánh giá độ chính xác của mô hình

score = model.evaluate(train\_generator,verbose=0)

print('Sai số kiểm tra là: ',score[0])

print('Độ chính xác kiểm tra là: ',score[1])

import matplotlib.pyplot as plt

# Draw plot

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epochs')

plt.legend(['train','Validation'])

plt.show()

plt.plot(history.history['loss'],'r',label='training loss')

plt.plot(history.history['val\_loss'],label='validation loss')

plt.xlabel('# epochs')

plt.ylabel('loss')

plt.legend()

plt.show()

from tensorflow.keras.models import load\_model

model.save('Gender2.h5')

model\_CNN = load\_model('Gender2.h5')

from tensorflow.keras.utils import load\_img, img\_to\_array

import numpy as np

filename = "/kaggle/input/gender/2.Female/18\_1\_0\_20170109212906609.jpg.chip.jpg"

predict = ['Male','Female']

predict = np.array(predict)

img = load\_img(filename,target\_size=(150,150))

plt.imshow(img)

img = img\_to\_array(img)

img = img.reshape(1,150,150,3)

img = img.astype('float32')

img = img/255

result = np.argmax(model\_CNN.predict(img),axis=-1)

predict[result]

***Real-time gender and face recognition***

import cv2 #thư viện xử lý hình ảnh và video trong Python. Nó cung cấp các chức năng để đọc, ghi và xử lý các hình ảnh từ các nguồn đầu vào khác nhau

import os

from tensorflow.keras.preprocessing.image import img\_to\_array

from tensorflow.keras.models import load\_model

import numpy as np

import cvlib #một thư viện xử lý hình ảnh dựa trên OpenCV, cung cấp các công cụ giúp phát hiện khuôn mặt, đồng thời cung cấp chức năng nhận biết giới tính và cảm xúc từ khuôn mặt

import tkinter as tk

from tkinter import \*

from PIL import ImageTk, Image

from tkinter import messagebox

import threading

# Load model

# face\_classifier = cv2.CascadeClassifier('face\_detection.xml')

gender\_model = load\_model('Gender1.h5')

emotion\_model = load\_model('Emotion1.h5')

gender\_labels = ['Male', 'Female']

emotion\_labels = ['Neutral', 'Happy', 'Sad', 'Surprised', 'Angry']

# Create tkinter window

root = tk.Tk()

root.geometry('1050x620')

root.resizable(False, False)

root.title('University of Economics Ho Chi Minh City')

icon = PhotoImage(file='img/Logo\_UEH\_xanh.png')

root.iconphoto(True, icon)

is\_running = False

def use\_camera():

    global is\_running

    is\_running = True

    start\_button.config(state="disabled")

    stop\_button.config(state="normal")

    exit\_button.config(state="normal")

    worker\_thread = threading.Thread(target=camera\_worker)

    worker\_thread.start()

def quit\_program():

    answer = messagebox.askyesno("Quit", "Do you want to exit?")

    if answer:

        root.destroy()

def cancel\_feed():

    global is\_running

    is\_running = False

    start\_button.config(state="normal")

    stop\_button.config(state="disabled")

def camera\_worker():

    capture = cv2.VideoCapture(0)

    while is\_running:

        ret, frame = capture.read()

        # Face detection

        faces, confidences = cvlib.detect\_face(frame)

        for face, confidence in zip(faces, confidences):

            # Get the coordinates of the face rectangle

            (startX, startY) = face[0], face[1]

            (endX, endY) = face[2], face[3]

            # Draw rectangle around the face

            cv2.rectangle(frame, (startX, startY), (endX, endY), (0, 0, 255), 2) #BGR

            # Crop the detected face region

            face\_crop = np.copy(frame[startY:endY, startX:endX])

            if face\_crop.shape[0] < 10 or face\_crop.shape[1] < 10:

                continue

            # Preprocess the face for gender prediction

            face\_crop = cv2.resize(face\_crop, (150, 150))

            face\_crop = face\_crop.astype("float") / 255.0

            face\_crop = img\_to\_array(face\_crop)

            face\_crop = np.expand\_dims(face\_crop, axis=0)

            # Predict gender

            conf\_model\_gender = gender\_model.predict(face\_crop)[0]

            idx\_model\_gender = np.argmax(conf\_model\_gender)

            label\_model\_gender = gender\_labels[idx\_model\_gender]

            # Predict emotion

            conf\_model\_emotion = emotion\_model.predict(face\_crop)[0]

            idx\_model\_emotion = np.argmax(conf\_model\_emotion)

            label\_model\_emotion = emotion\_labels[idx\_model\_emotion]

            label = "{},{}".format(label\_model\_gender, label\_model\_emotion)

            Y = startY - 10 if startY - 10 > 10 else startY + 10

            # Write the predicted gender label on the image

            cv2.putText(frame, label, (startX, Y), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (255, 255, 0), 2)

        # Convert the image from OpenCV BGR format to PIL Image

        image = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB))

        image = image.resize((640, 480), Image.ANTIALIAS)

        # Convert the PIL Image to ImageTk to display on Tkinter label

        imgtk = ImageTk.PhotoImage(image=image)

        # Update the image on the label

        image\_label.configure(image=imgtk)

        image\_label.image = imgtk

        if cv2.waitKey(1) & 0xFF == ord('s'):

            break

    capture.release()

    cv2.destroyAllWindows()

# Main frame

main\_frame = tk.Frame(root, bg='#faf79b')

main\_frame.pack(side=tk.LEFT)

main\_frame.pack\_propagate(False)

main\_frame.configure(width=1050, height=620)

# Title 1

label\_title = tk.Label(main\_frame, text='PREDICT GENDER AND EMOTION BASED ON HUMAN FACES',

                       font=("Arial", 20),

                       fg="red",

                       bg='#faf79b')

# Title 2

label\_title2 = tk.Label(main\_frame, text='Artificial Intelligent in Business',

                        font=("Arial", 15),

                        fg="blue",

                        bg='#faf79b')

# Title 3

label\_title3 = tk.Label(main\_frame, text='Le Thi Huyen',

                        font=("Arial", 15),

                        fg="blue",

                        bg='#faf79b')

# Camera frame

image\_label = tk.Label(main\_frame, bg='#D9EAF4')

image\_label.place(x=160, y=110, width=750, height=450)

# Start button

start\_button = tk.Button(main\_frame,

                         text="START",

                         font=('Bold', 15),

                         fg='white',

                         bd=0,

                         bg='blue',

                         command=use\_camera)

start\_button.place(x=300, y=570, width=80, height=35)

# Stop button

stop\_button = tk.Button(main\_frame, text="STOP",

                        font=('Bold', 15),

                        fg='white',

                        bd=0,

                        bg='blue',

                        command=cancel\_feed,

                        state="disabled")

stop\_button.place(x=500, y=570, width=80, height=35)

# Exit button

exit\_button = tk.Button(main\_frame, text="EXIT",

                        font=('Bold', 15),

                        fg='white',

                        bd=0,

                        bg='blue',

                        command=quit\_program,

                        state="normal")

exit\_button.place(x=700, y=570, width=80, height=35)

label\_title.pack()

label\_title2.pack()

label\_title3.pack()

root.mainloop()