

# 2023年【科學探究競賽-這樣教我就懂】

## 普高組 成果報告表單

**題目名稱：** 利用AI演算法以及心理學知識於智慧眼鏡

### 一、摘要

利用人工智慧演算法和心理學知識應用於智慧眼鏡的開發，旨在根據被觀察者的表現，辨識他的情緒狀況。此技術可應用於多種場合，作者使用公開資料，透過深度學習演算法提取多種面部表情及小動作，分析出各情緒的差異，建立模型並進行驗證。實驗結果顯示，該智慧眼鏡在情緒辨識上具有高準確度，可為實際應用提供有力的技術支持。未來，研究者將進一步優化模型，提升智慧眼鏡的實用性和應用範圍，為社會帶來更多的價值。

### 二、探究題目與動機

我們在日常生活中是否時常不知道對話者的情緒，很常惹怒對方卻不自知，大家時常會因為察覺不出對話者的情緒，而造成吵架、溝通不良、被欺騙等的結局，所以我們設計了這個題目AI演算法以及心理學知識於智慧眼鏡，我們想要探究此智慧眼鏡能不能有效的提升人與人之間溝通的效率和有效的改溝通的隔閡。

### 三、探究目的與假設

(一)、智慧眼鏡能否幫忙輔助判斷對方的情緒

### 四、探究方法與驗證步驟

(一)、蒐集關於解讀情緒的心理學資料

我們申請到中國科學院微表情數據庫的CAS(ME)<sup>3</sup> 數據庫( J. Li et al., "CAS(ME)<sup>3</sup> : A Third Generation Facial Spontaneous Micro-Expression Database with Depth Information and High Ecological Validity," )，拿到裡面微表情的視頻資料庫。包含7種情感類型：Happiness (64 samples), Disgust (281), Fear(93), Anger(70), Sadness(64), Surprise (201) and Others (170)。

微表情出現時間介於時間在1/25 秒至 1/5 秒，宏表情持續時間在 1/5 秒以上。

(二)、利用 python 建立機器學習模型

```
import cv2
import os
import numpy as np
from sklearn.preprocessing import LabelEncoder
```

```

from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,
TimeDistributed, LSTM
from tensorflow.keras.callbacks import ModelCheckpoint
from keras.preprocessing.image import img_to_array

# 定義數據路徑和標籤
data_path = "path/to/dataset"
labels = ["Happiness", "Disgust", "Fear", "Anger", "Sadness", "Surprise", "Others"]

# 初始化數據和標籤
data = []
target = []

# 定義表情標籤
EMOTIONS = ['Happiness', 'Disgust', 'Fear', 'Anger', 'Sadness', 'Surprise', 'Others']

# 定義模型
model = cv2.dnn.readNetFromCaffe("deploy.prototxt.txt",
"res10_300x300_ssd_iter_140000.caffemodel")

# 定義計數器和閾值
micro_frames = 0
macro_frames = 0
micro_threshold = 30 * 1/25 # 微表情閾值 · 1/25秒
macro_threshold = 30 * 1/5 # 宏表情閾值 · 1/5秒

# 定義視頻捕獲器
cap = cv2.VideoCapture('video.mp4')

# 循環遍歷視頻幀
while True:

```

```

ret, frame = cap.read()
if not ret:
    break

# 將幀轉換為 blob

(h, w) = frame.shape[:2]
blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 1.0, (300, 300), (104.0, 177.0,
123.0))

# 將 blob 送入模型中進行前向推理

model.setInput(blob)
detections = model.forward()

# 遍歷檢測結果

for i in range(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]

    # 過濾掉置信度低的檢測結果

    if confidence < 0.5:
        continue

    # 提取人臉區域的坐標

    box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
    (startX, startY, endX, endY) = box.astype("int")

    # 提取人臉區域並進行情緒分類

    face = frame[startY:endY, startX:endX]
    face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
    face = cv2.resize(face, (48, 48))
    face = face.astype("float") / 255.0
    face = img_to_array(face)
    face = np.expand_dims(face, axis=0)
    preds = model.predict(face)[0]
    label = EMOTIONS[preds.argmax()]

```

```

# 根據表情持續時間分類為微表情或宏表情
if macro_frames >= macro_threshold:
    print("宏表情: " + label)
    macro_frames = 0
elif micro_frames >= micro_threshold:
    print("微表情: " + label)
    micro_frames = 0
else:
    micro_frames += 1

# 遍歷數據路徑中的每一個視頻文件
for label in labels:
    label_path = os.path.join(data_path, label)
    for filename in os.listdir(label_path):
        file_path = os.path.join(label_path, filename)

        # 讀取視頻文件並將每一幀轉換為灰度圖像
        cap = cv2.VideoCapture(file_path)
        frames = []
        while True:
            ret, frame = cap.read()
            if not ret:
                break
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            frames.append(gray)

        # 將每個視頻轉換為一組幀的序列
        sequence = np.stack(frames, axis=0)

        # 將標籤添加到目標列表中
        target.append(label)

```

```

# 將序列添加到數據列表中

data.append(sequence)

# 將數據和標籤轉換為NumPy數組
data = np.array(data)
target = np.array(target)

# 將標籤進行編碼
le = LabelEncoder()
target = le.fit_transform(target)
target = to_categorical(target)

# 打亂數據集
indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
target = target[indices]

# 將數據集分為訓練集和測試集
split = int(0.8 * data.shape[0])
train_data = data[:split]
train_target = target[:split]
test_data = data[split:]
test_target = target[split:]

# 定義模型
model = Sequential()
model.add(TimeDistributed(Conv2D(32, (3, 3), activation="relu"), input_shape=(None, 120, 120, 1)))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2, 2))))
model.add(TimeDistributed(Conv2D(64, (3, 3), activation="relu")))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2, 2))))
model.add(TimeDistributed(Conv2D(128, (3, 3), activation="relu")))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2, 2))))

```

```

model.add(TimeDistributed(Flatten()))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(32, activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(7, activation="softmax"))

# 編譯模型
model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])

# 定義回調函數，保存最佳模型
checkpoint = ModelCheckpoint("best_model.h5", monitor="val_accuracy", save_best_only=True,
mode="max")

# 訓練模型
model.fit(train_data, train_target, epochs=10, batch_size=32)

```

### (三)、用PyTorch 訓練模型

1. 導入CAS(ME)<sup>3</sup> 數據庫並進行訓練。

### (四)、用卷積神經網路 (CNN) 進行模型優化

### (五)、輸出結果

用test\_data實驗得到以下結果：

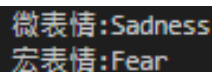


圖1 測試結果

## 五、結論與生活應用

我們利用心理學知識及AI演算法，訓練一個能判斷人們的情緒的模型，我們把它應用在了眼鏡上，我們稱這個眼鏡為智慧眼鏡。同樣的他可以應用在警察審問犯人、法官在判斷被告人是否擁有悔改表現、跟人交流的時候可以用來判斷是否自己惹怒了對方.....等。

## 參考資料

吳奇、申尋兵、傅小蘭. (2010). 微表情研究及其應用. 心理科學進展, 18(9), 1359-1368.

J. Li et al., "CAS(ME)3: A Third Generation Facial Spontaneous Micro-Expression Database With Depth Information and High Ecological Validity," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 2782-2800, 1 March 2023, doi: 10.1109/TPAMI.2022.3174895.