

ĐỒ ÁN CUỐI KÌ

ĐỀ TÀI: NHẬN DIỆN NGỦ GẬT

GVHD: PGS.TS Nguyễn Trường Thịnh
SVTH: Đỗ Công Danh
MSSV: 19146163

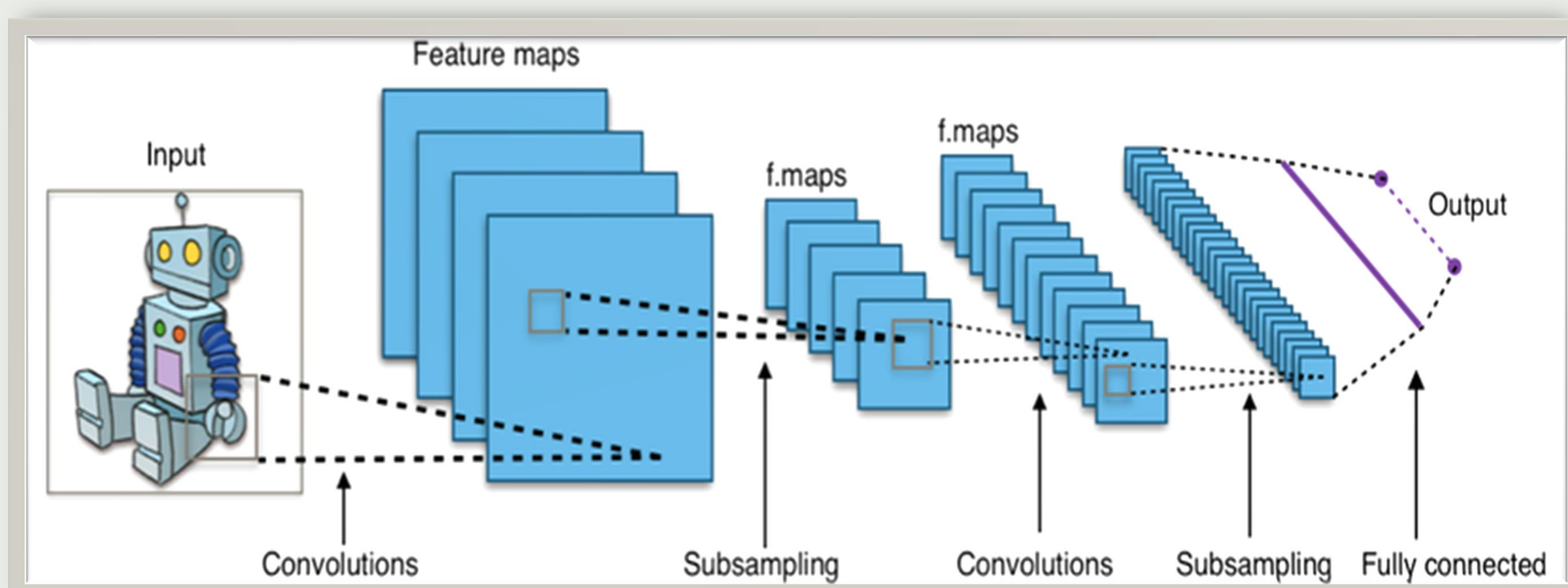
GIỚI THIỆU

Báo cáo về "Rối loạn giấc ngủ và tai nạn giao thông" tại hội nghị khoa học thường niên Hội Hô hấp Việt Nam và Chương trình đào tạo y khoa liên tục 2015, giáo sư Telfilo Lee Chiong (Trung tâm National Jewish Health, Mỹ), cho biết buồn ngủ là một trong những nguyên nhân chính gây tai nạn giao thông trên thế giới. Ước tính khoảng 10-15% tai nạn xe có liên quan đến thiếu ngủ. Nghiên cứu về giấc ngủ ở các tài xế 19 quốc gia châu Âu cho thấy tỷ lệ buồn ngủ khi lái xe cao, trung bình 17%.



HƯỚNG GIẢI QUYẾT

Sử dụng mô hình cấu trúc của mạng CNN



Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như Relu và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

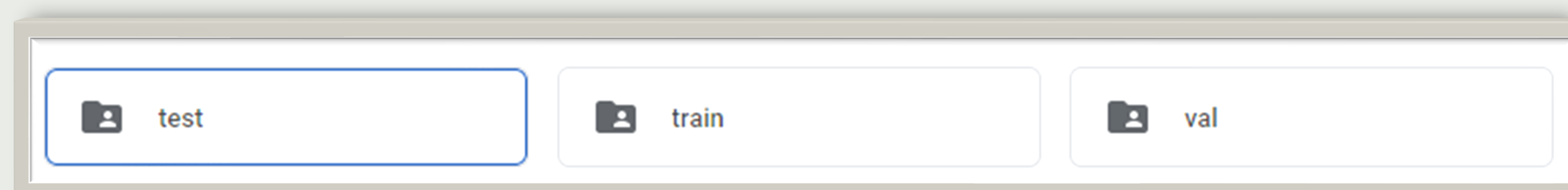
THÔNG TIN LIÊN HỆ

SVTH: Đỗ Công Danh
SDT: 0374426066
Gmail:
19146163@student.hcmute.edu.vn
Link github:
<https://github.com/DoDanh/Project-AI>
Mã QR:



THỰC NGHIỆM

Tạo dữ liệu



Tạo mô hình để chạy

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense
model=Sequential()

model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',input_shape=(150,150,3)))
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D((2,2)))

model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D((2,2)))

model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D((2,2)))

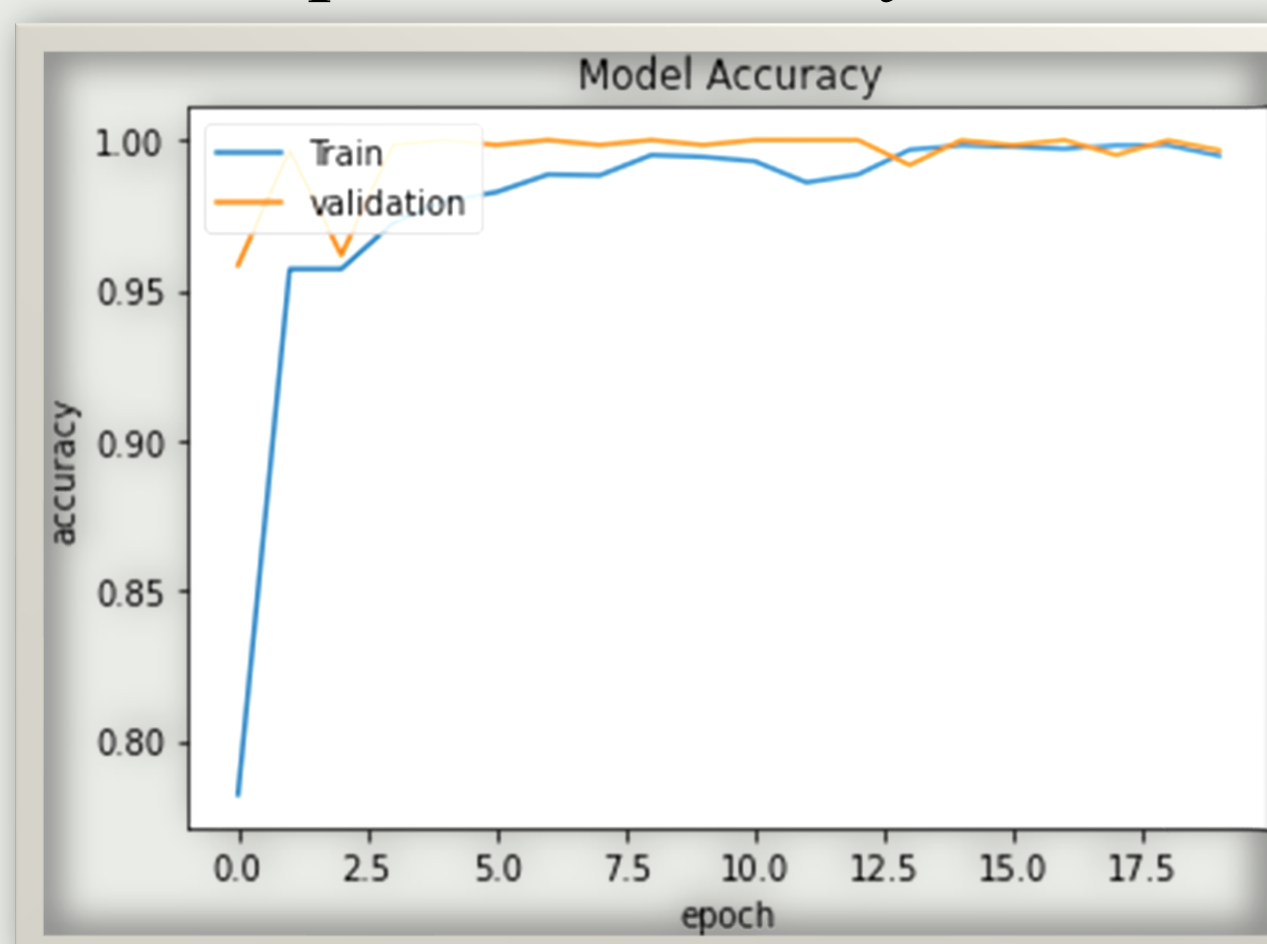
model.add(Flatten())
model.add(Dense(128,activation='relu',kernel_initializer='he_uniform'))
model.add(Dense(2,activation='softmax'))
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 150, 150, 32)	896
conv2d_1 (Conv2D)	(None, 150, 150, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_2 (Conv2D)	(None, 75, 75, 64)	18496
conv2d_3 (Conv2D)	(None, 75, 75, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_4 (Conv2D)	(None, 37, 37, 128)	73856
conv2d_5 (Conv2D)	(None, 37, 37, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 128)	0
flatten (Flatten)	(None, 41472)	0
dense (Dense)	(None, 128)	5308544
dense_1 (Dense)	(None, 2)	258

Total params: 5,595,810
Trainable params: 5,595,810
Non-trainable params: 0

KẾT QUẢ THU ĐƯỢC

Kết quả sau khi chạy mô hình :



Độ chính xác khi chạy mô hình:

```
Score=model.evaluate(training_set,verbose=0)
print('Train Loss', Score[0])
print('Train Accuracy', Score[1])
```

Train Loss 0.01867879368364811
Train Accuracy 0.9941176176071167

Thử nghiệm video realtime



Thử nghiệm với hình ảnh

