

MARMARA UNIVERSITY
FACULTY OF ENGINEERING
ELECTRICAL-ELECTRONICS

EE4069: Artificial Intelligence in Health Sciences



NAME

STUDENT NUMBER

ÖMER BUMİNHAN ÇAY 150718004

Submitted to: Gökhan Bora Esmer

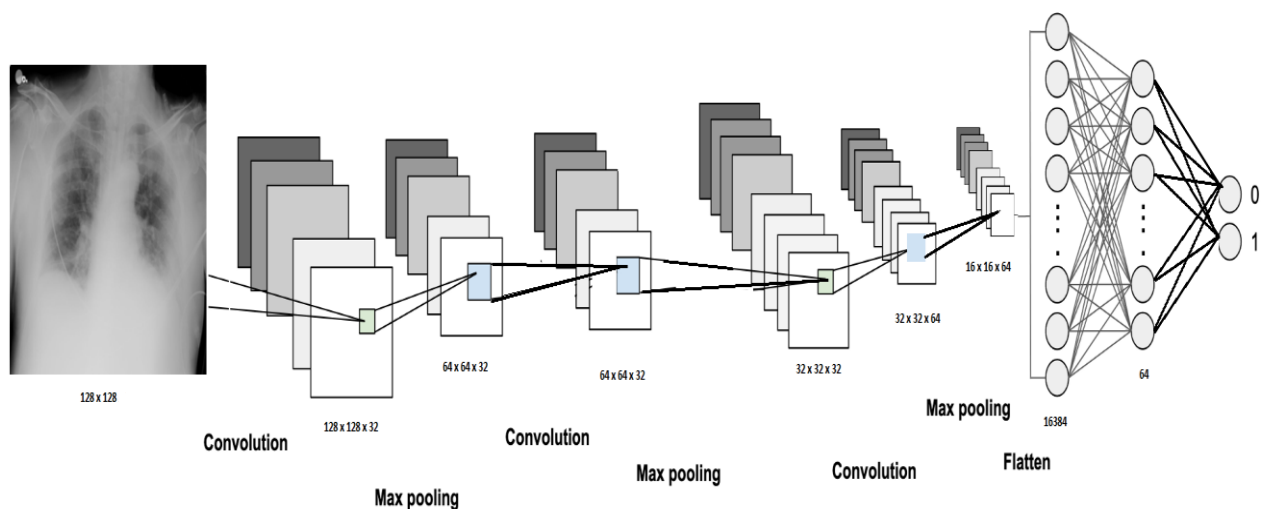
Due Date: 09/01/2023

Introduction

In this project, I made an algorithm that classify the CT images as binary whether there is covid 19 or not.

My Algorithm

Schematic image of the model for the image classifier:



I choose this model because CNN is very efficient way to make a binary classification. Firstly, with the Convolutional layers model recognized the features of the images. After that I used the Max Pooling layer because this layer down scale the images and it summarizes the features. Also, with the Dropout layer I tried to prevent the overfitting. The last part of my model is having just 2 different layers. I used Flatten layer because I just make 2-D arrays to 1 linear vector. After Flatten layer I used Dense layer for connecting my neurons and decreasing my neuron number. I used the activation modes all 'relu' except the last layer. I choose the 'relu' because it gives faster and more accurate results. For the last layer I choose the 'softmax' activation mode because it is an efficient way for binary classification. After that I choose my loss function is 'Sparse Categorical Cross Entropy' because it is faster and more suitable for the classification.

Model code:

```
model = Sequential([
    Conv2D(32, 3, activation='relu', padding='same', input_shape=(128, 128, 3)),
    MaxPooling2D(),
    Conv2D(32, 3, activation='relu', padding='same'),
    MaxPooling2D(),
    Conv2D(64, 3, activation='relu', padding='same'),
    MaxPooling2D(),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(2, activation='softmax')
])
```

```

MaxPooling2D(2, padding='same'),
Dropout(0.3),

Conv2D(32, 3, activation='relu', padding='same'),
MaxPooling2D(2, padding='same'),
Dropout(0.5),

Conv2D(64, 3, activation='relu', padding='same'),
MaxPooling2D(2, padding='same'),
Dropout(0.2),

Flatten(),
Dense(64, activation='relu'),
Dense(2, activation='softmax')

])
model.compile(optimizer = 'adam', loss = 'SparseCategoricalCrossentropy', metrics = ['accuracy'])
model.summary()

```

Output for the model code:

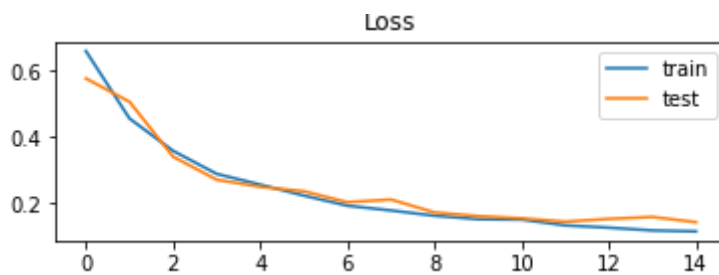
Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 128, 128, 32)	896
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
dropout (Dropout)	(None, 64, 64, 32)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout_1 (Dropout)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_2 (Dropout)	(None, 16, 16, 64)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 64)	1048640
dense_1 (Dense)	(None, 2)	130
=====		
Total params: 1,077,410		
Trainable params: 1,077,410		
Non-trainable params: 0		
=====		

I trained the model with this code lines and arranged epoch 15 and batch size 128.

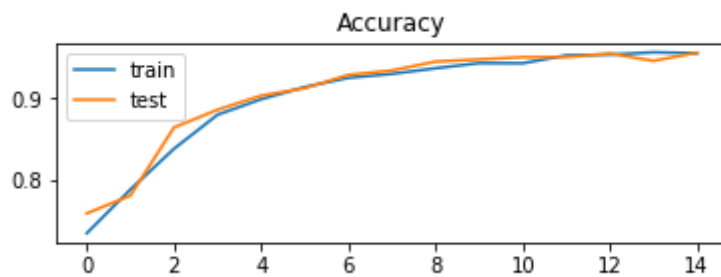
```
history = model.fit(x_train, y_train, epochs = 15, batch_size = 128, v  
alidation_data = (x_val, y_val))
```

Results

Training loss and validation loss plot:



Training accuracy and the validation accuracy plot:



Finally, I calculated the test data accuracy with this code:

```
c = 0  
for i in range(len(y_test)):  
    a = np.expand_dims(x_test[i],axis=0)  
    pred = model.predict(a)  
    result = 0  
    if (pred[0,0]<pred[0,1]):  
        result = 1  
    if result==y_test[i]:  
        c = c+1  
  
print(c/len(y_test))
```

Test data accuracy score is:

0.9558291093410572

Conclusion

In this project I learned how layer works and how to make a binary classification for the images. Also, what should I know about the code for the image classifier. My model is working well, I think. Bu it can be improved with different models, and it can make more accurate predicts than my model.