



TUGAS 2

Implementasi Jaringan Saraf Tiruan

Ahmad Faqih Al Ghiffary

2108107010072

Table of Contents

1	2	3	4	5
Jenis Kasus	Dataset	Fitur	Label	Jenis Jaringan Saraf Tiruan
6	7	8	9	10
Jenis Optimisasi	Fungsi Aktivasi	Jumlah Hidden Layer	Total Hidden Node/Layer	Total Bobot (Weight)

Jenis Kasus & Link Dataset

Jenis kasus yang saya gunakan pada proyek ini yaitu mengklasifikasi gambar mata dengan kelopak terbuka dan kelopak tertutup.

Link: <https://www.kaggle.com/datasets/charunisa/eyes-dataset>.

Dataset memiliki total 4850 gambar.



Data Explorer

Version 1 (2.14 MB)

- dataset
 - train
 - closed
 - open
 - val
 - closed
 - open

Summary

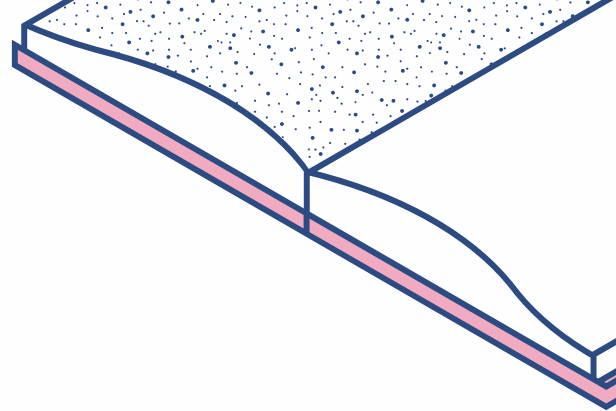
- 4850 files



Jumlah Fitur

Fitur dalam dataset yang saya pilih gambar ialah jumlah pixel dalam gambar tersebut. Gambar pada dataset berukuran 24x24 yaitu 576 fitur.

Image Type	jpeg (JPEG)
Width	24 pixels
Height	24 pixels



Jumlah Label

Label adalah kategori atau klasifikasi yang akan diprediksi oleh model. Terdapat 2 label yang terdapat pada dataset saya yaitu **closed** dan **open**.

Label **closed** berisi gambar mata tertutup.

Label **open** berisi gambar mata terbuka.

```
▼ TUGAS2AI
  ▼ data
    ▼ testing
      > closed
      > open
    ▼ training
      > closed
      > open
```



Jenis Saraf Jaringan


Saya menggunakan Jenis Jaringan Saraf Tiruan **Convolutional Neural Network (CNN)**. Saya menggunakan CNN dikarenakan CNN merupakan salah satu jenis jaringan saraf tiruan yang utama digunakan untuk dataset gambar.



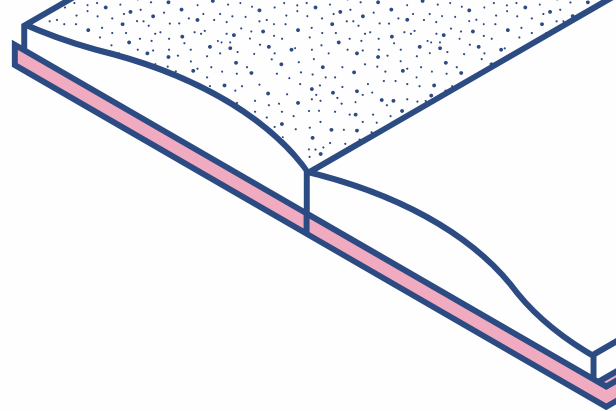


Jenis Optimisasi

Saya menggunakan jenis optimisasi bernama **RMSprop** (**Root Mean Squared Propagation**). dengan learning rate bernilai **1e-4** (0.0001).



```
1 model.compile(  
2     loss='binary_crossentropy',  
3     optimizer=RMSprop(learning_rate=1e-4),  
4     metrics=['accuracy']  
5 )
```



Fungsi Aktivasi

Saya menggunakan dua jenis fungsi aktivasi:

- **ReLu (Rectified Linear Unit)** $f(x) = \max(0, x)$

Fungsi ini didefinisikan sebagai

- **Sigmoid** $f(x) = \frac{1}{1 + e^{-x}}$

Fungsi ini didefinisikan sebagai

```
1 model = tf.keras.models.Sequential([
2     tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)), # 3 = RGB
3     tf.keras.layers.MaxPooling2D(2, 2), # Pooling Layer
4     tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
5     tf.keras.layers.MaxPooling2D(2,2), # Pooling Layer
6     tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
7     tf.keras.layers.MaxPooling2D(2,2), # Pooling Layer
8     tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
9     tf.keras.layers.MaxPooling2D(2,2), # Pooling Layer
10
11     # from 2D to 1D (Flatten)
12     tf.keras.layers.Flatten(),
13     tf.keras.layers.Dropout(0.5),
14     tf.keras.layers.Dense(512, activation='relu'),
15     tf.keras.layers.Dense(1, activation='sigmoid')
16 ])
```


Jumlah Hidden Layer

conv2d (Conv2D)	[Input Layer]
max_pooling2d (MaxPooling2D)	
conv2d_1 (Conv2D)	[Hidden Layer]
max_pooling2d_1 (MaxPooling2D)	
conv2d_2 (Conv2D)	[Hidden Layer]
max_pooling2d_2 (MaxPooling2D)	
conv2d_3 (Conv2D)	[Hidden Layer]
max_pooling2d_3 (MaxPooling2D)	
flatten (Flatten)	[Layer Klasifikasi]
dropout (Dropout)	
dense (Dense)	

Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 128)	0
flatten (Flatten)	(None, 6272)	0
dropout (Dropout)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776

Total Hidden Node/Layer

Pada kolom **Output Shape**, (None, a, a, X)

Nilai **X** menunjukkan banyak node pada layer tersebut

Total = $32+32+64+64+128+128+128+128=704$

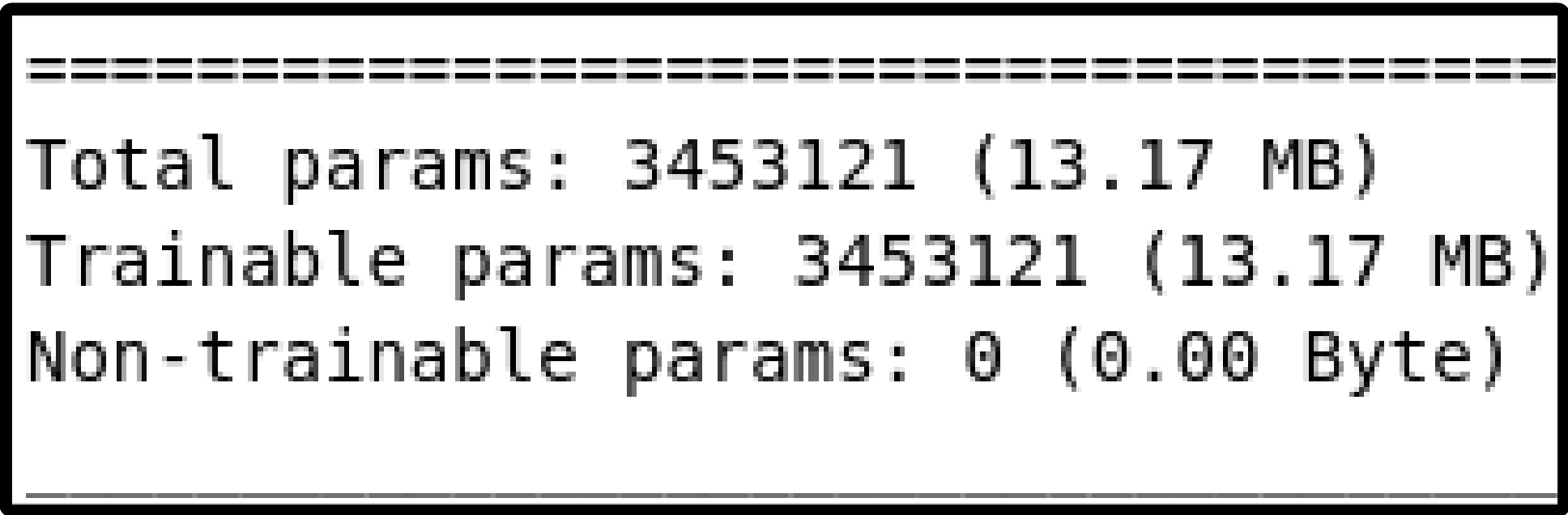
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 128)	0
flatten (Flatten)	(None, 6272)	0
dropout (Dropout)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776



Jumlah Total Bobot (Weight)

Menggunakan `model.summary()`, kita dapat melihat jumlah bobot pada model tersebut. Dalam kasus ini, jumlah bobot bernilai 3.453.121



```
=====
Total params: 3453121 (13.17 MB)
Trainable params: 3453121 (13.17 MB)
Non-trainable params: 0 (0.00 Byte)
```



Sekian &
Terima Kasih