

# Laporan Workshop Deep Learning Cenim 2020



Disusun Oleh:

Muhammad Azki Rasyid

07211840000072

Departemen Teknik Komputer  
Fakultas Teknologi Elektro Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

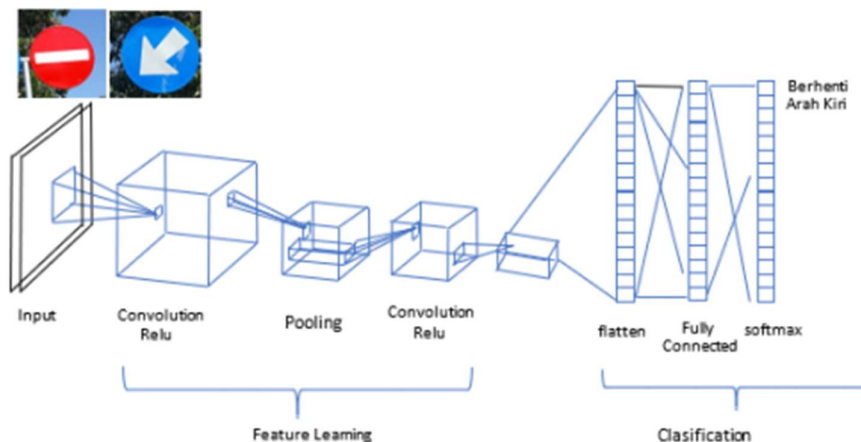
Dalam pengenalan citra berbasis deep learning menggunakan keras dan backend tensor flow terdapat *Autonomous Vehicle* yang berupa Autonomous Car dengan fitur bantuan yang ditawarkan untuk mengambil alih kemudi saat berkendara. Dalam Autonomous Vehicle ini terdapat beberapa tantangan seperti pengguna jalan lain, kondisi cuaca, kondisi infrastruktur, kondisi lalu lintas, dan rambu-rambu lalu lintas. Autonomous Vehicle ini terdapat Passive sensor dengan resolusi tinggi dengan warna, frame rate yang konstan, biaya yang murah, dan citra dapat dengan mudah dimengerti. Dalam mendeteksi object pada citra, terdapat beberapa permasalahan seperti perbedaan luminasi, dan perbedaan orientasi dan skala. Dalam pengenalan citra ini tentukan menggunakan deep learning.

Pada Multi Layer Perceptron terdapat lima tahap dalam membuat program prediksi data, yaitu:

1. Menyiapkan Data set,
2. Mendefinisikan Model MLP,
3. Compile Model Keras,
4. Fitting Model MLP Ke Data Set,
5. Membuat Prediksi.

Convolutional Neural Network

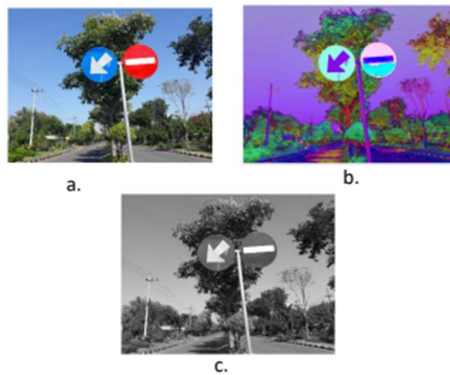
Berikut merupakan Arsitektur dari Convolutional Neural Network:



Gambar 3.1 Contoh Arsitektur Convolutional Neural Network

Dalam Convolutional Network terdapat input berupa citra:

1. Citra Berwarna RGB
2. Citra HSV
3. Citra Gray



Gambar 3.2 Data input (a) RGB, (HSV) dan (c) Gray scale

## Pooling Layer

Pooling Layer bertujuan untuk mengurangi jumlah parameter Ketika citra terlalu besar dengan cara untuk mengurangi dimensi setiap fitur, Pooling digunakan untuk memperkecil ukuran, namun jumlah fiturnya tetap sama. Maxpooling merupakan teknik untuk mengurangi dimensi dengan cara mengambil nilai dengan elemen yang terbesar sesuai dengan ukuran filter. Sebagai contoh pada gambar dibawah adalah maxpooling dengan filter 2X2 dan stride 2. Stride sama dengan jarak. Jadi stride 2 artinya antar kotak sekarang ke kotak lain berjarak 2.



Klasifikasi :

Flatten → data dijadikan 2 dimensi, informasi yang berada pada flatten merupakan fitur

Maxpooling:

Mengurangi dimensi dengan cara mengambil nilai dengan elemen yang terbesar sesuai dengan ukuran filter.

Sebagai contoh pada Gambar dibawah adalah max polling dengan filter 2x2 dan stride dua.

1. Langkah pertama yaitu mengimport library dari python

```
1 import os
2 from keras.models import load_model
3 import cv2
4 import numpy as np
5 from keras.layers import Input, Dense, Conv2D, MaxPooling2D, Flatten
6 from keras.models import Model
```

Library yang digunakan yaitu os, numpy, input, dense, conv2d, maxpooling2d, flatten, dan model

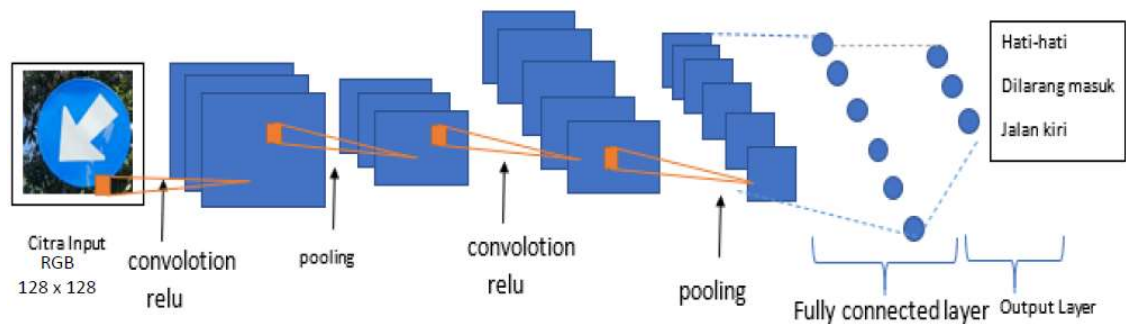
2. Langkah kedua yaitu ,embuat fungsi untuk me-load gambar

```

9 def LoadCitraTraining(sDir,LabelKelas,TargetKelas):
10     X=[]
11     D=[]
12     for i in range(len(LabelKelas)):
13         DirKelas = os.path.join(sDir, LabelKelas[i])
14         files = os.listdir(DirKelas)
15         for f in files:
16             ff=f.lower()
17             print(f)
18             if (ff.endswith('.jpg')|ff.endswith('.jpeg')|ff.endswith('.png')):
19                 NmFile = os.path.join(DirKelas,f)
20                 img= cv2.imread(NmFile,1)
21                 img=cv2.resize(img,(128,128));
22                 img= np.asarray(img)
23                 img= np.asarray(img)/256
24                 img=img.astype('float32')
25                 X.append(img)
26                 D.append(TargetKelas[i])
27     X=np.array(X)
28     D=np.array(D)
29     X=X.astype('float32')
30     D=D.astype('float32')
31     return X,D

```

3. Langkah ketiga yaitu membuat model CNN



Gambar arsitektur model CNN yang ingin dibangun

```

51 def ModelDeepLearningCNN(JumlahKelas):
52     input_img = Input(shape=(128, 128, 3))
53     x = Conv2D(32, (3, 3), activation='relu', padding='same')(input_img)
54     x = MaxPooling2D((2, 2), padding='same')(x)
55     x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
56     x = MaxPooling2D((2, 2), padding='same')(x)
57     x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
58     x = Flatten()(x)
59     x = Dense(100,activation='relu')(x)
60     x = Dense(100,activation='relu')(x)
61     x=Dense(JumlahKelas,activation='softmax')(x)
62     ModelCNN = Model(input_img, x)
63     ModelCNN.compile(loss='categorical_crossentropy', optimizer='adam',
64                       metrics=['accuracy'])
65     return ModelCNN

```

4. Langkah selanjutnya membuat fungsi training atau fit model ke input

```
75 def TrainingCNN(DirektoriDataSet,LabelKelas,TargetKelas,JumlahKelas,  
76                 NamaFileModel,bTrainingBaru):  
77     X,D=LoadCitraTraining(sDir,LabelKelas,TargetKelas)  
78     if bTrainingBaru==1:  
79         ModelCNN =ModelDeepLearningCNN(JumlahKelas)  
80     else:  
81         ModelCNN = load_model(NamaFileModel)  
82     ModelCNN.fit(X, D,epochs=100,shuffle=True)  
83     ModelCNN.save(NamaFileModel)  
84     return ModelCNN
```

Fungsi training ini merupakan fungsi dimana model akan di fit dengan data input X dan D. Setelah di fit atau ditraining, model ini disimpan untuk nantinya digunakan lagi ke data yang berbeda selain dengan training

5. Langkah selanjutnya yaitu membuat fungsi prediksi dan anotasi

```
62 def PrediksiCitra(img,ModelCNN):  
63     im =cv2.resize(img,(128,128))  
64     v=np.array(im)  
65     v=v.astype('float32')  
66     v=v/255  
67     hs=ModelCNN.predict(v.reshape(1,128,128,3))  
68     return hs  
69  
70 def LabelAnotasi(TargetKelas,hs):  
71     if hs.max()>0.8:  
72         idx = np.max(np.where( hs == hs.max()))  
73     else:  
74         idx=-1  
75     return idx
```

6. Langkah selanjutnya menjalankan fungsi yang sudah digunakan

```

import os
from ModulKlasifikasiCitra import *

LabelKelas = [];
LabelKelas.append("Belok Kiri")
LabelKelas.append("Dilarang Masuk")
LabelKelas.append("Hati Hati")

TargetKelas=([1,0,0],
               [0,1,0],
               [0,0,1])

dirpath = os.getcwd('D:\Kuliah\Semester 5\PCV\Deep Learn
sDir=dirpath+'Dilarang Masuk'
md = load_model('weight.h5')
img= cv2.imread(sDir+"20200821-123521-176992.jpg",1)
hs=PrediksiCitra(img,md)

idx= LabelAnotasi(TargetKelas,hs)
if idx>-1:
    print(LabelKelas[idx])
else:
    print("None")

cv2.imshow('ImageOri',img)

cv2.waitKey()
cv2.destroyAllWindows()

```

Fungsi yang ditulis akan diimport di file lain. Setelah directory ditentukan, dataset langsung dijalankan