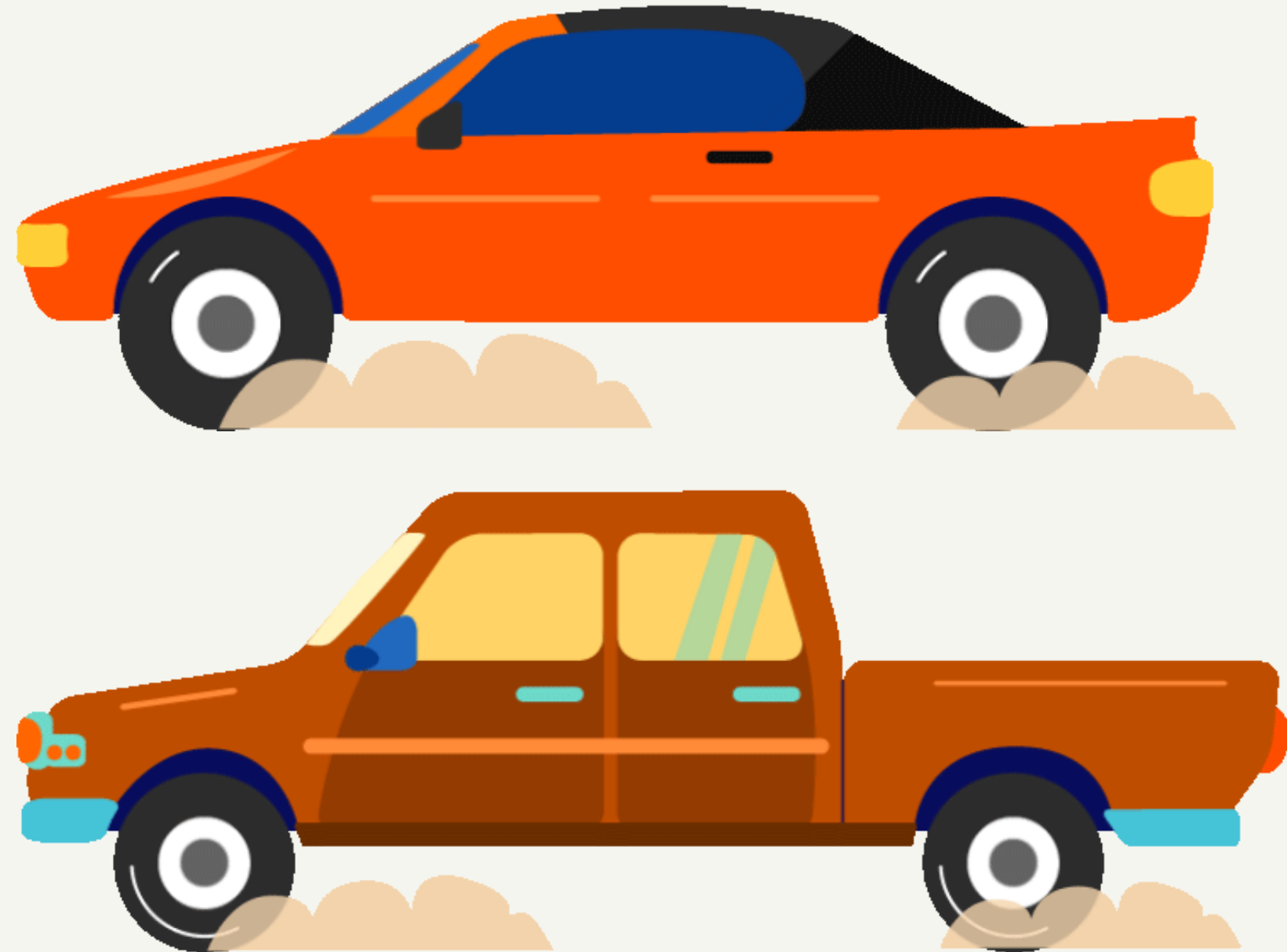


# Toll Fare Vehicle Classification using CNN

Axel Lie  
Leonardo Ignatius



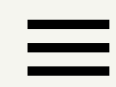
— 01





# Introduction

## Part 01



# Background

Golongan kendaraan yang melintas di jalan tol, dibedakan menurut jumlah gandar atau as roda.

Pengklasifikasian ini bertujuan untuk memisahkan tarif tol.

Gerbang tol yang kini ada terbagi menjadi dua jenis.

Marketing & Communication Department Head Jasa Marga menjelaskan, sistem pemilahan golongan masih manual.

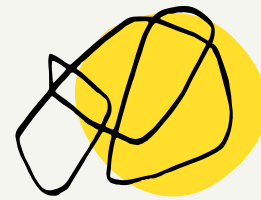
Untuk gardu multi golongan (I-V) penentuan golongan kendaraan dilakukan oleh petugas di dalam control room.

Cara tersebut dilakukan, sehingga ketika pengemudi melakukan tapping uang elektroniknya, tarif yang keluar akan sesuai dengan golongan jenis kendaraannya.





# The Problem



— 05

What we want to solve

Kami ingin mengklasifikasikan setiap kendaraan ke dalam golongannya masing-masing (I-V) melalui gambar kendaraan.



# Methods

## Part 02





# Overview of Research Methods

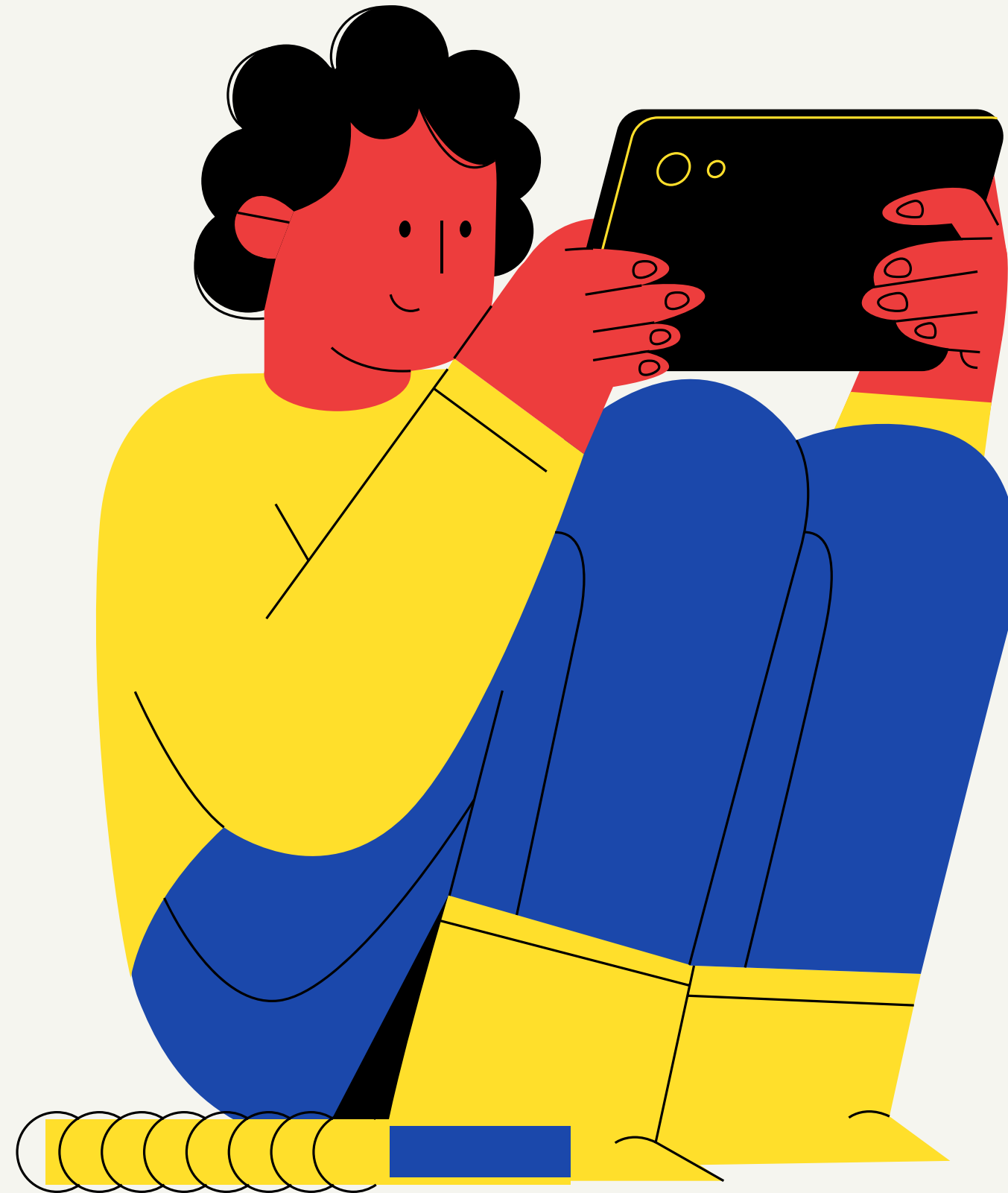
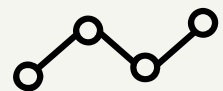
Convolutional Neural Network (CNN) adalah salah satu jenis neural network yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image.

Secara garis besar Convolutional Neural Network (CNN) tidak jauh beda dengan neural network biasanya. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Convolutional layer juga terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels).



# Dataset

Kami membuat datasetnya sendiri dengan cara mengambil gambar kendaraan tiap golongan(I-V) dari google images sebanyak 50 gambar tiap golongan. Jadi totalnya ada 250 gambar.





# Results

## Part 03





# Results



```
1 model = tf.keras.models.Sequential([
2     tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(150, 200, 3)),
3     tf.keras.layers.MaxPooling2D(3, 3),
4     tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
5     tf.keras.layers.MaxPooling2D(3,3),
6     tf.keras.layers.Conv2D(256, (3,3), activation='relu'),
7     tf.keras.layers.MaxPooling2D(3,3),
8     tf.keras.layers.Flatten(),
9     tf.keras.layers.Dense(512, activation='relu'),
10    tf.keras.layers.Dropout(0.1),
11    tf.keras.layers.Dense(256, activation='relu'),
12    tf.keras.layers.Dropout(0.1),
13    tf.keras.layers.Dense(128, activation='relu'),
14    tf.keras.layers.Dropout(0.1),
15    tf.keras.layers.Dense(64, activation='relu'),
16    tf.keras.layers.Dropout(0.1),
17    tf.keras.layers.Dense(5, activation='softmax')
18 ])
```

— 14

## Accuracy

Training Accuracy : 94%

Validation Accuracy : 57%

Epoch 30/30

30/30 [=====] - 11s 372ms/step - loss: 0.1470 - accuracy: 0.9459 - val\_loss: 1.7425 - val\_accuracy: 0.5750  
<tensorflow.python.keras.callbacks.History at 0x7f30a117ce90>

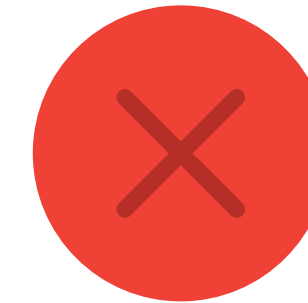


# Predict Results



```
1 img = image.load_img('/content/Mobil/Golongan 3/golongan3 (28).jpg', target_size=(150,200))
2 imgplot = plt.imshow(img)
3 x = image.img_to_array(img)
4 x = np.expand_dims(x, axis=0)
5 images = np.vstack([x])
6 classes = model.predict(images, batch_size=10)
7 predictclass(classes)
```

↳ Golongan 4



```
1 img = image.load_img('/content/Mobil/Golongan 1/golongan1 (1).jpg', target_size=(150,200))
2 imgplot = plt.imshow(img)
3 x = image.img_to_array(img)
4 x = np.expand_dims(x, axis=0)
5 images = np.vstack([x])
6 classes = model.predict(images, batch_size=10)
7 predictclass(classes)
```

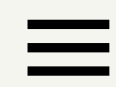
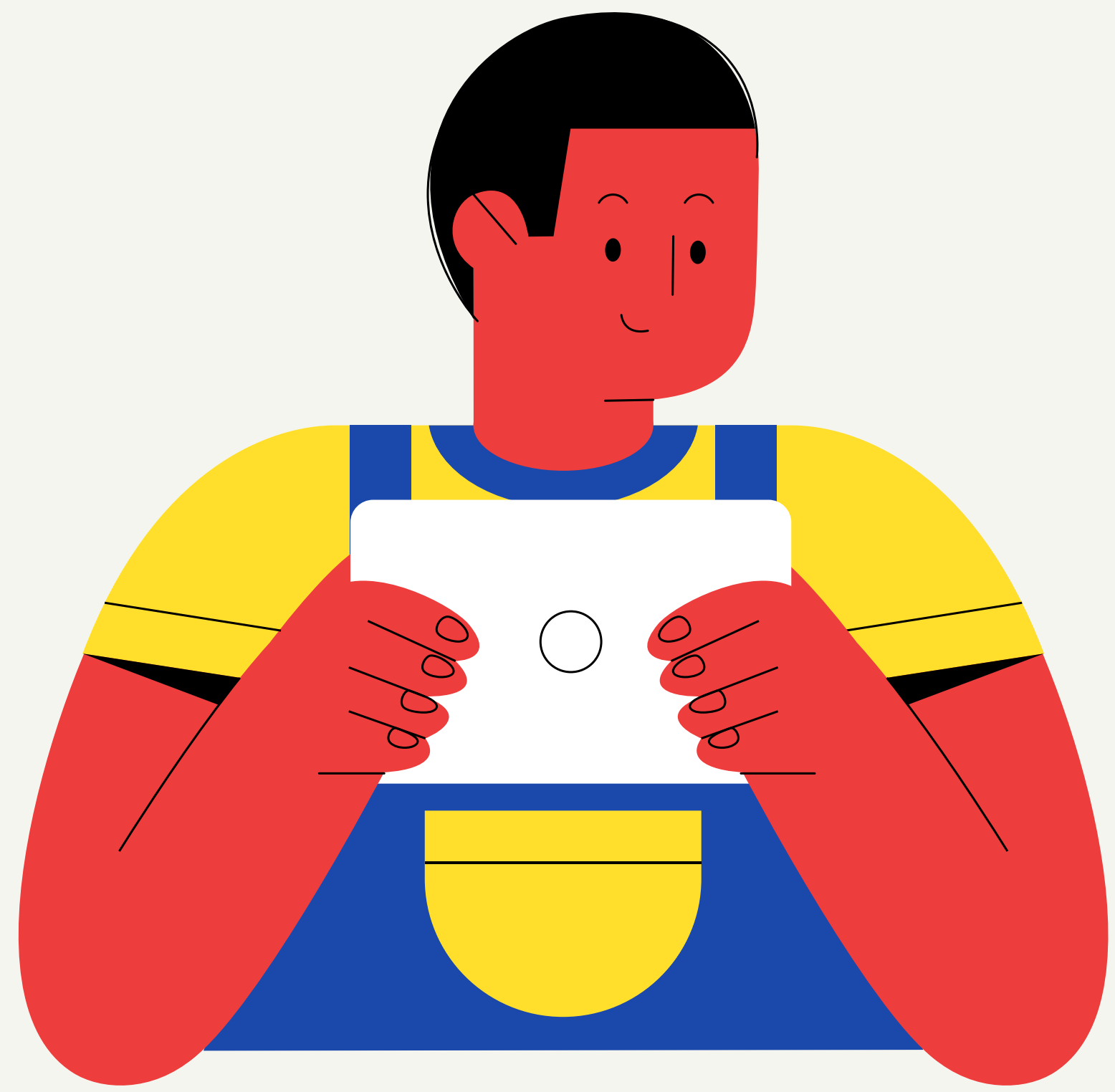
↳ Golongan 1





# Conclusion

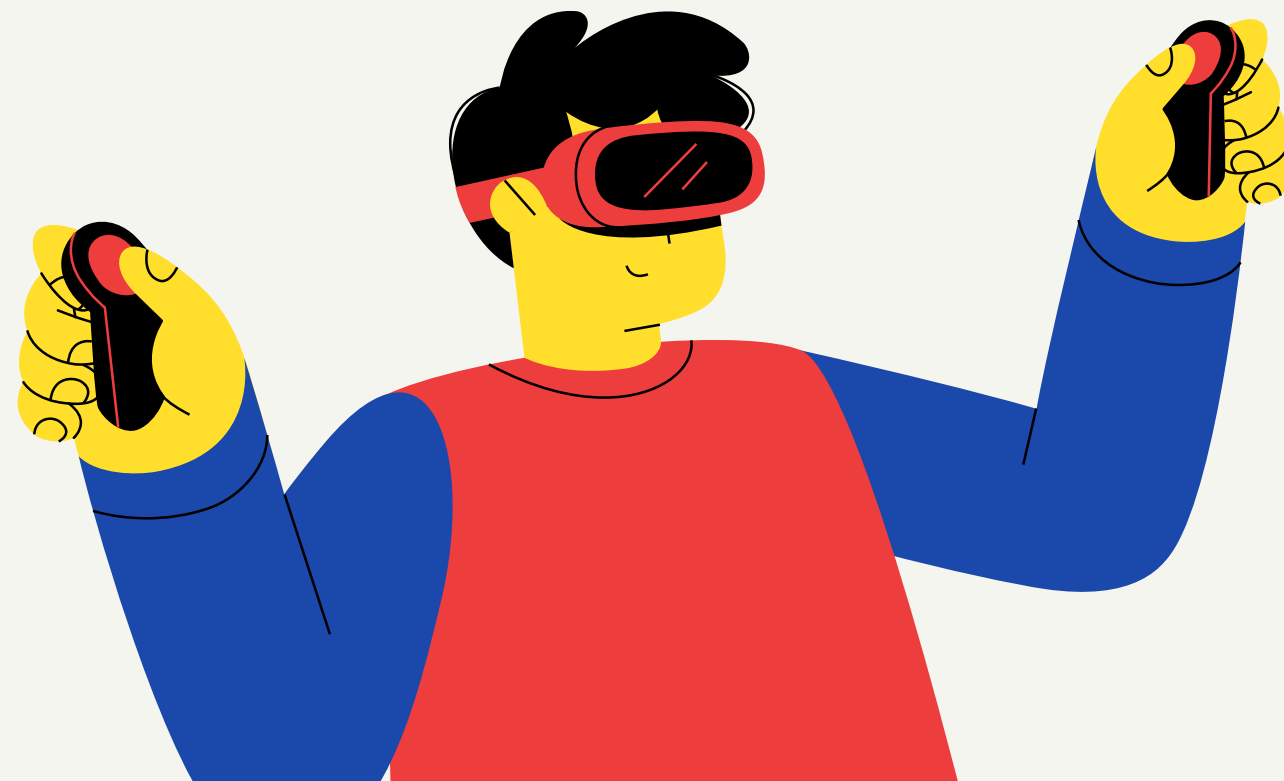
## Part 04



# Conclusion

Model sudah mengklasifikasi beberapa gambar dengan benar, namun masih ada beberapa gambar yang dipredict dengan salah. Dan juga Validation Accuracy masih kecil. Hal ini dikarenakan beberapa hal :

1. Angle dari tiap gambar kendaraan masih berbeda
2. Jumlah dataset kurang banyak





Thank you