

# Digital Talent Scholarship 2022

## Convolution Neural Network Pt 1

Lead a sprint through the Associate Machine Learning  
Track

# Agenda

- Computer Vision
- Convolution Neural Network
- Classifying Real-World Images
- Horse or Human
- Cats and Dogs

# Are your students ML-ready?

## Recap

**Apa itu Computer Vision?**

## Coding Session

mnist / fashion\_mnist dataset

```
mnist = tf.keras.datasets.fashion_mnist
```

# Computer Vision

```
import tensorflow as tf
```

```
mnist = tf.keras.datasets.fashion_mnist
```

```
(training_images, training_labels) , (test_images, test_labels) =  
mnist.load_data()
```

```
training_images = training_images/255.0
```

```
test_images = test_images/255.0
```

```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(),  
                                    tf.keras.layers.Dense(512, activation=tf.nn.relu),  
                                    tf.keras.layers.Dense(10, activation=tf.nn.softmax)])
```

```
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy')
```

```
model.fit(training_images, training_labels, epochs=5)
```

```
model.evaluate(test_images, test_labels)
```

```
classifications = model.predict(test_images)
```

```
print(classifications[0])
```

```
print(test_labels[0])
```

## Additional Codes

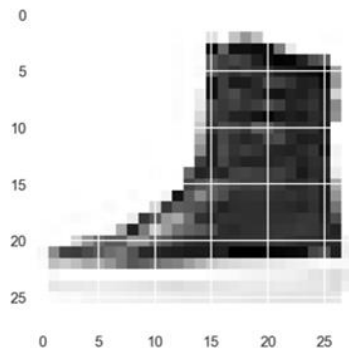
```
import matplotlib.pyplot as plt  
  
plt.imshow(training_images[0])  
  
print(training_labels[0])  
  
print(training_images[0])
```

# Limitations of DNN

1. Not 28x28 greyscale
2. Not Left-facing boot
3. Must be centered, well defined, similar shapes

**Solusinya? CNN!**

y = 7 (Sneaker)

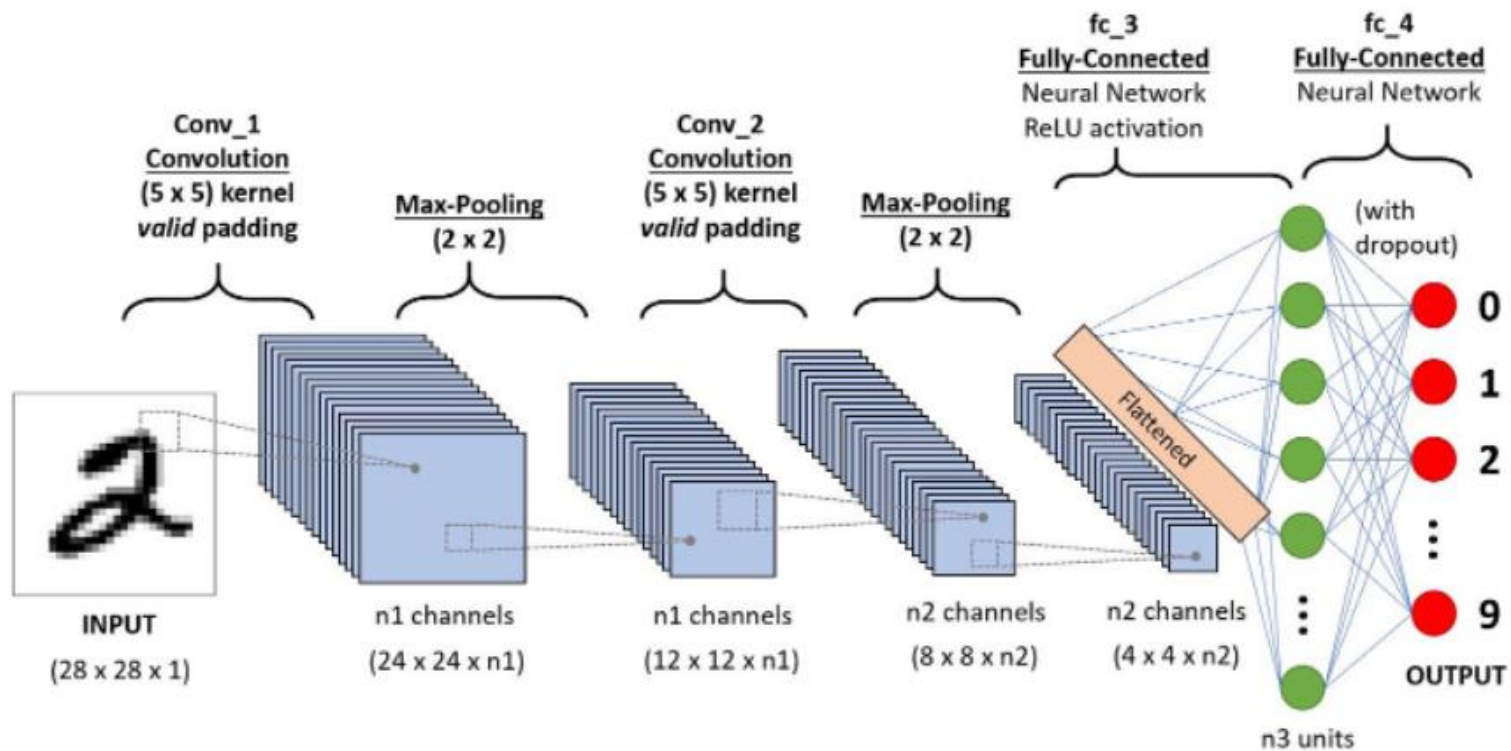




# Apa itu CNN

CNN atau bisa disebut sebagai **Convolution Neural Network** merupakan bagian dari AI

CNN merupakan bagian dari **Deep Learning**

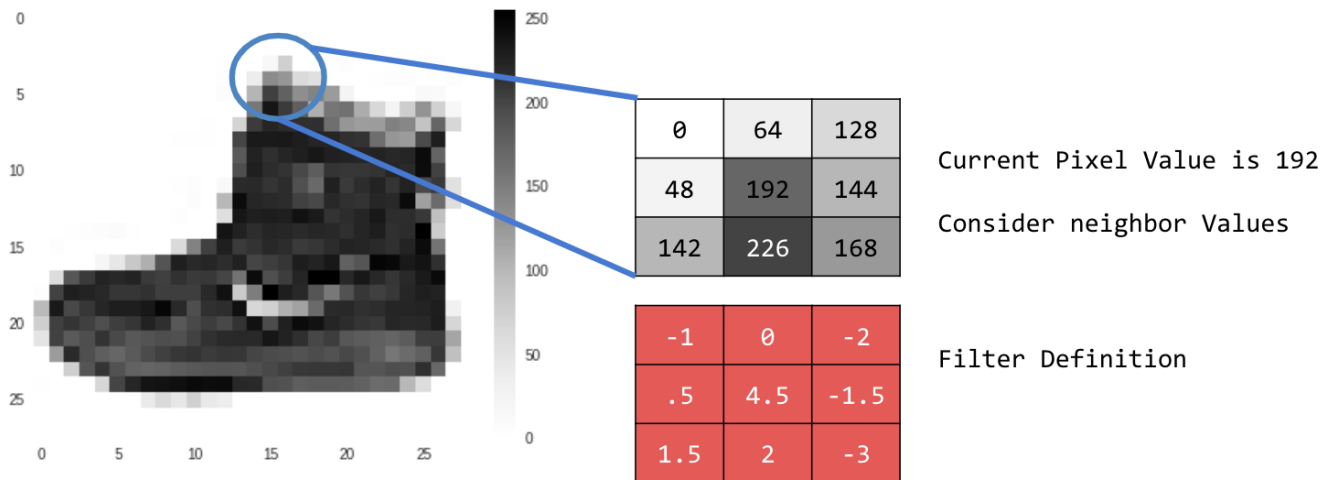


# Convolution NN

1. Mengekstrak fitur dalam sebuah gambar
2. Mencocokkan fitur yang tepat dengan set

**Bagian manakah dari proses ini yang merupakan Machine Learning?**

# Convolution NN



$$\begin{aligned} \text{CURRENT\_PIXEL\_VALUE} &= 192 \\ \text{NEW\_PIXEL\_VALUE} &= (-1 * 0) + (0 * 64) + (-2 * 128) + \\ &\quad (.5 * 48) + (4.5 * 192) + (-1.5 * 144) + \\ &\quad (1.5 * 42) + (2 * 226) + (-3 * 168) \end{aligned}$$

`tf.keras.layers.Conv2D(64, (3,3), activation='relu')`

# Convolutions is about Filter !

Input image

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Filter

1	0	1
0	1	0
1	0	1

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

# Convolution NN



-1	0	1
-2	0	2
-1	0	1



# Convolution NN

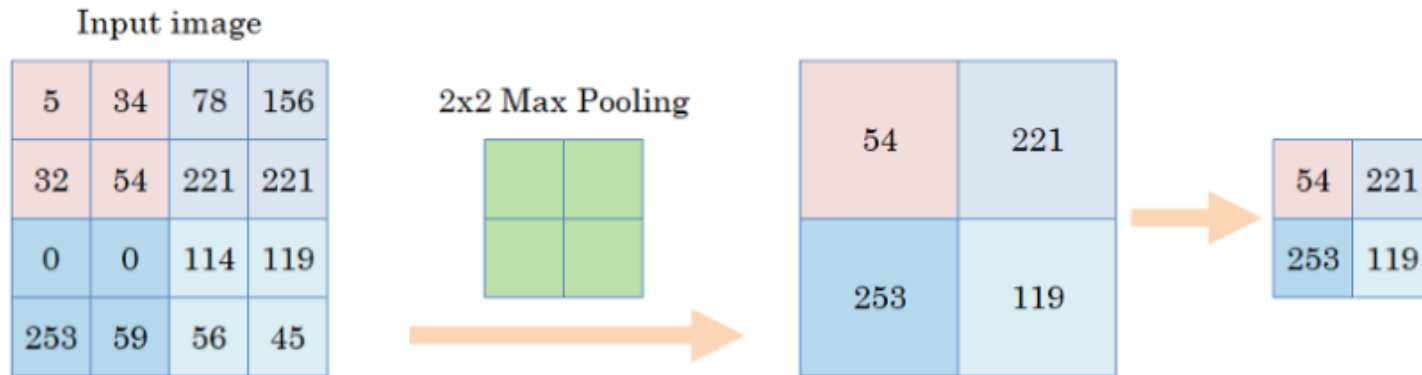


-1	-2	-1
0	0	0
1	2	1



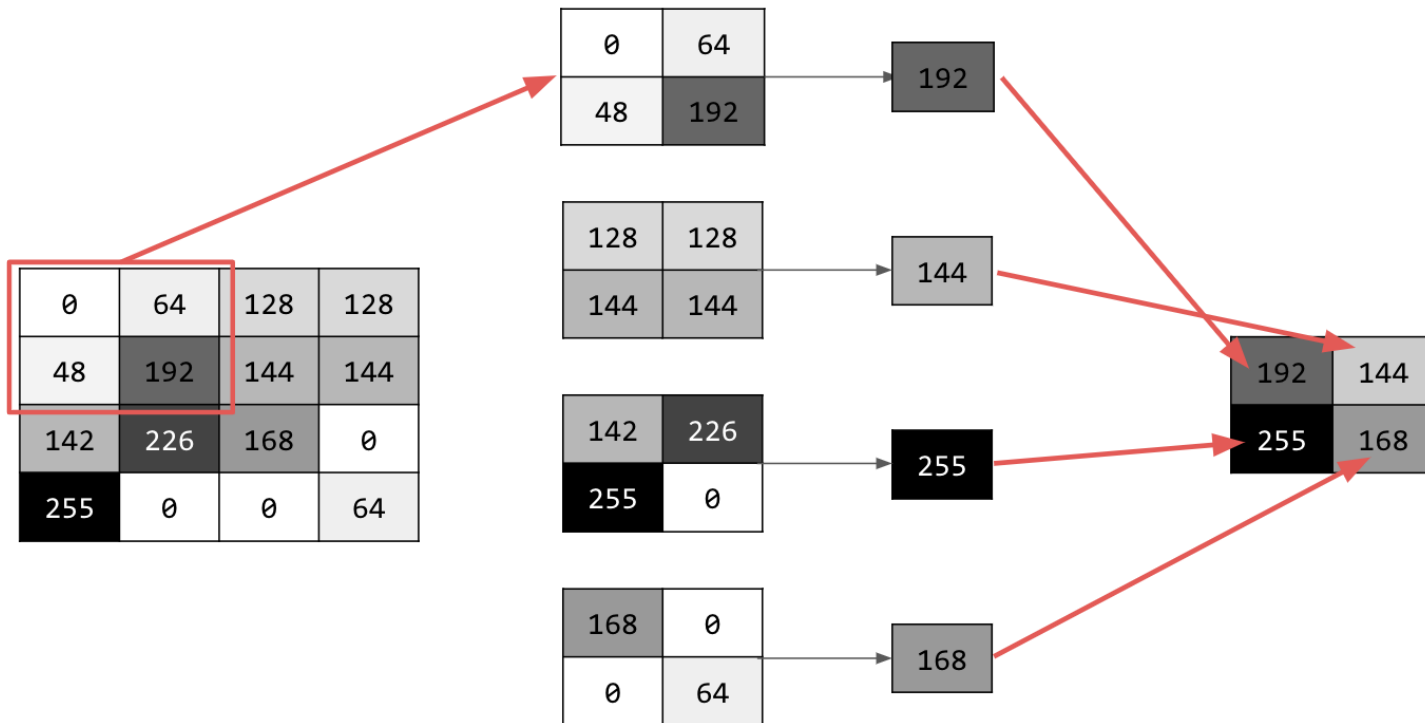


# Lalu Apa itu Pooling ?

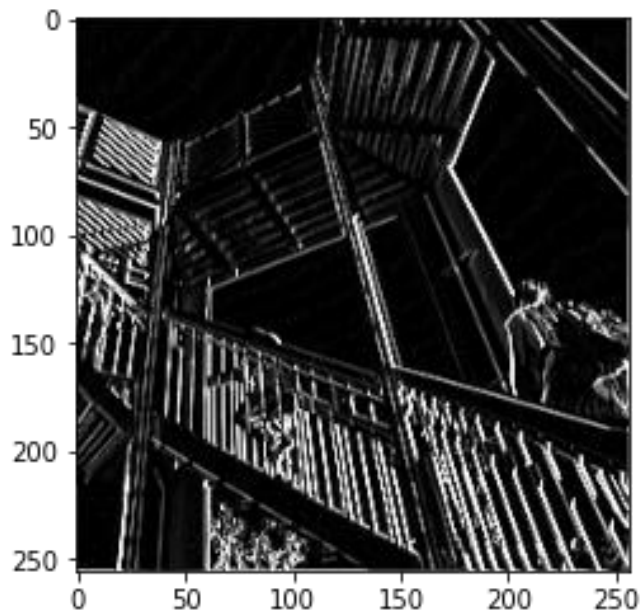
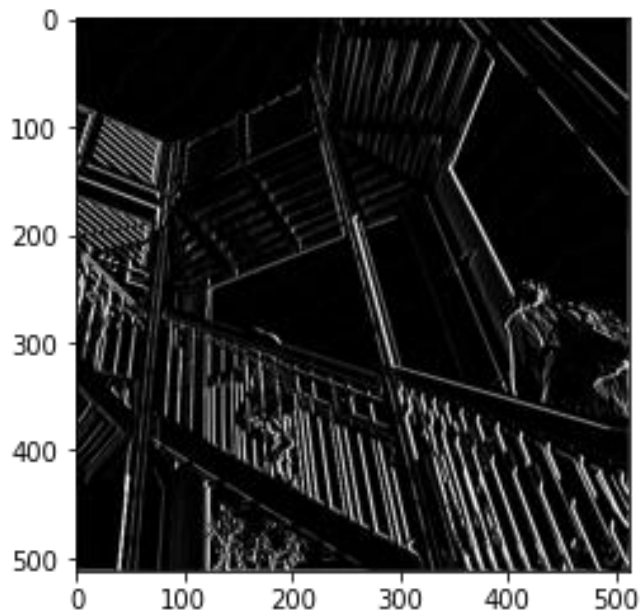


`tf.keras.layers.MaxPooling2D(2, 2)`

# Pooling



# Pooling



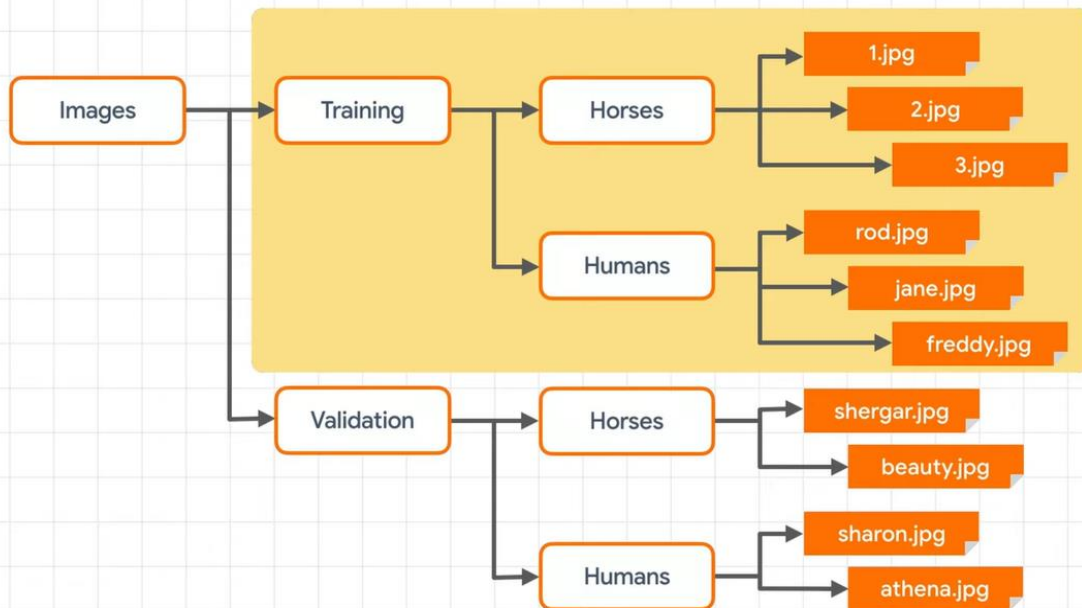
## Contoh Filter dan Pooling

<https://colab.research.google.com/drive/1CTxGUcjxQzvTWVFfF5j2LcopK82jN8E0>

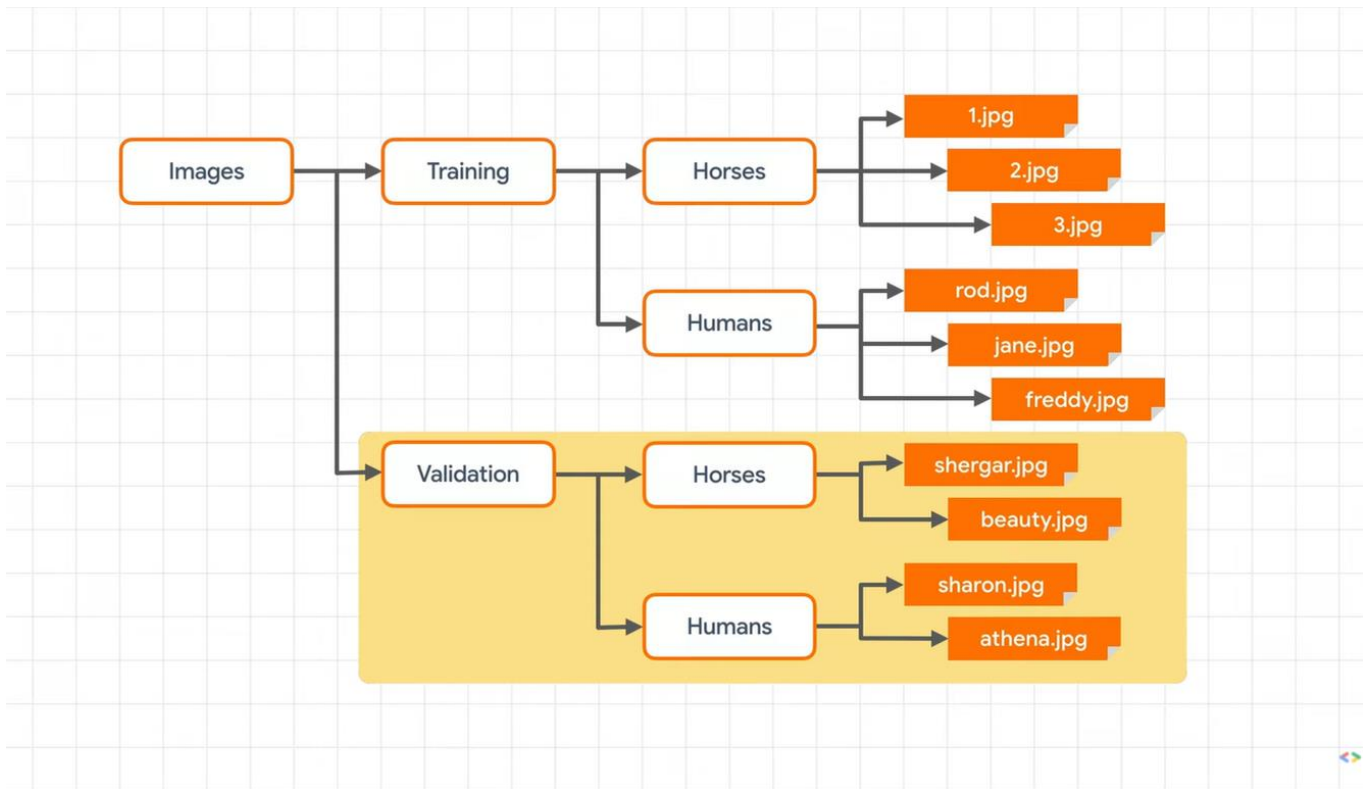
# Classifying Real-World Images



# Classifying Real-World Images



# Classifying Real-World Images



## Data Generator

- You can then use this to do some transformers in the image so just normalizing them
- in this case, this will be the training directory that contains the horses and humans sub directories.
- You only have two classes like we do here keep this is binary, if you have more classes it should be categorical feel validation datasets.
- These two generators now provide the images that your model can use for training and validation.



# Data Generator

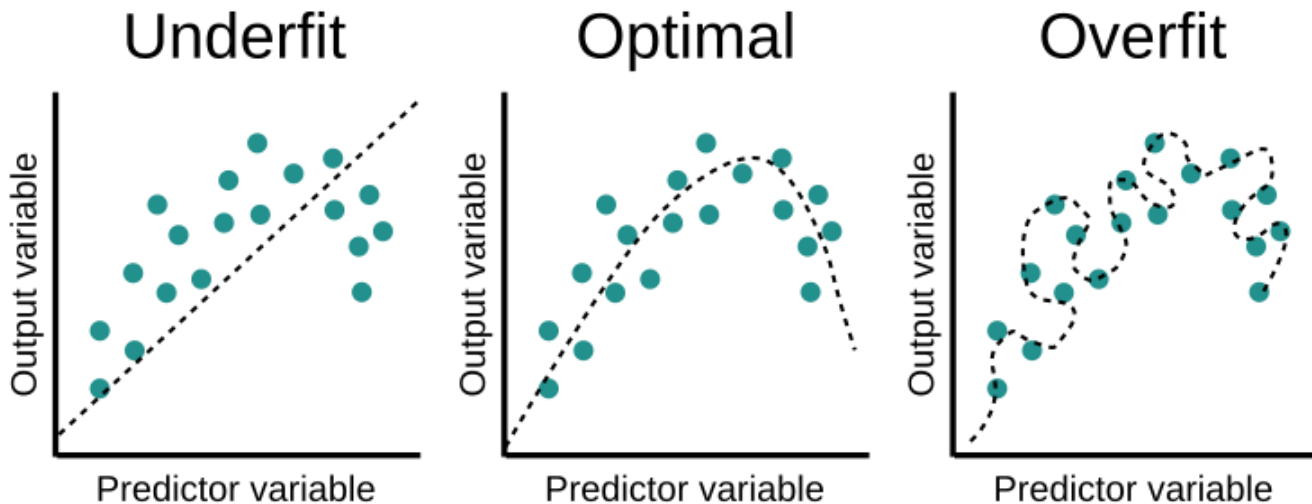
```
test_datagen = ImageDataGenerator(rescale=1./255)

validation_generator = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(300, 300),
    batch_size=32,
    class_mode='binary')
```

# Image Augmentation and Overfitting

Apa itu overfitting

Bagaimana cara menangani overfitting?



# Augmenting Images

```
# Updated to do image augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```





# Augmenting Images

```
# Updated to do image augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```



WIDTH SHIFTED

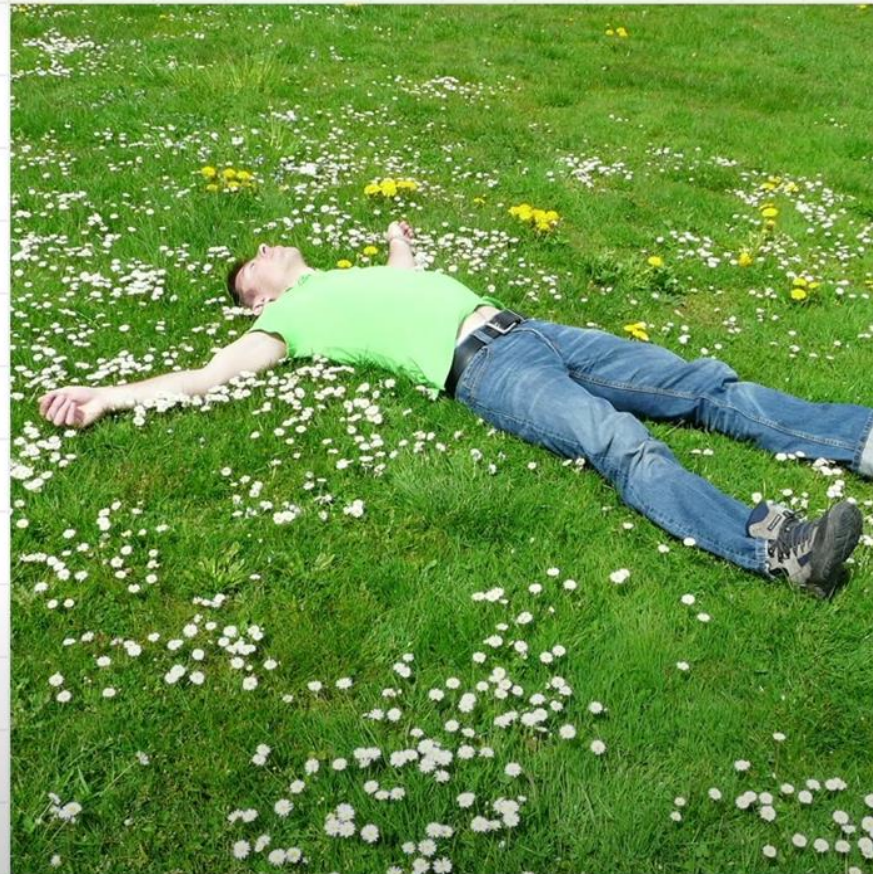
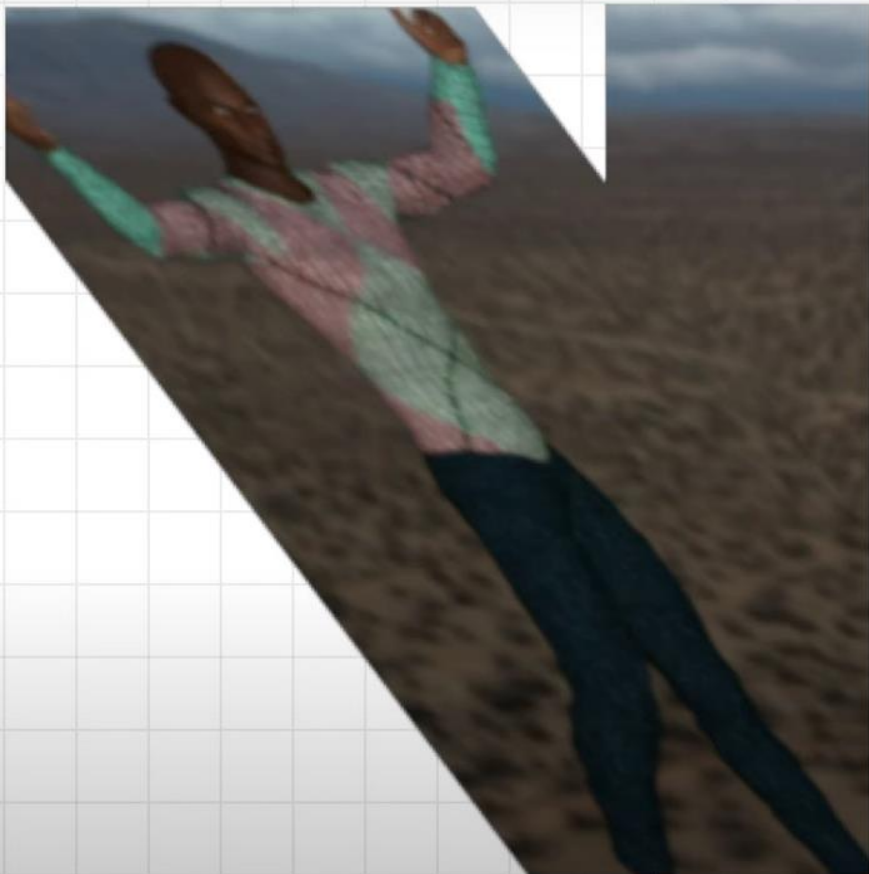


# Augmenting Images

```
# Updated to do image augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```



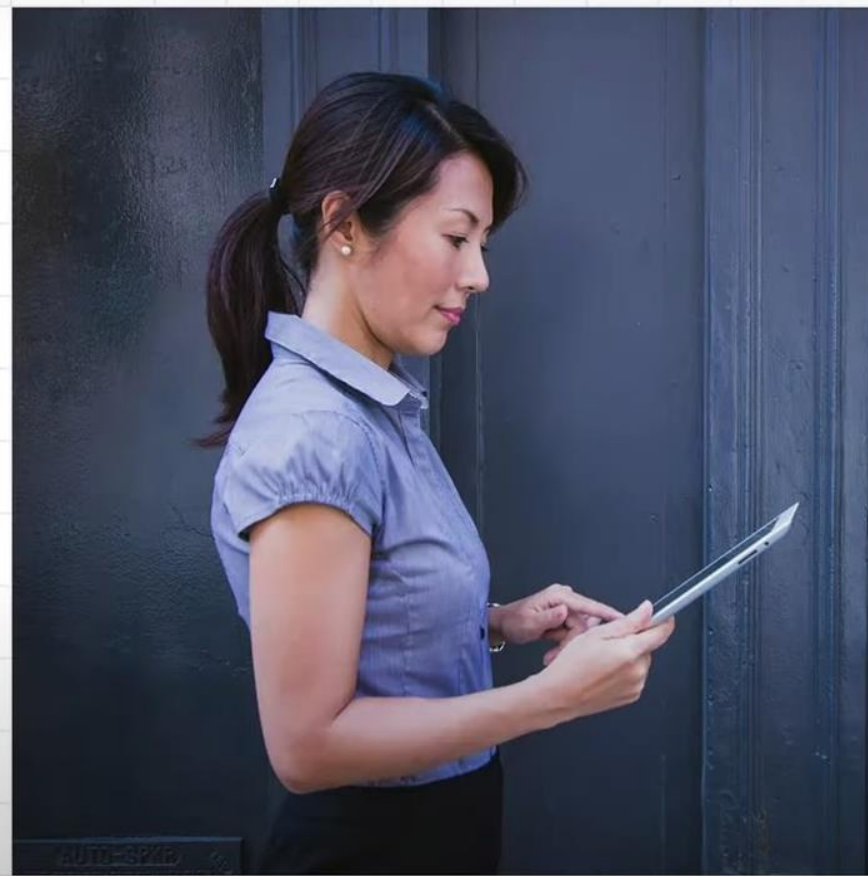


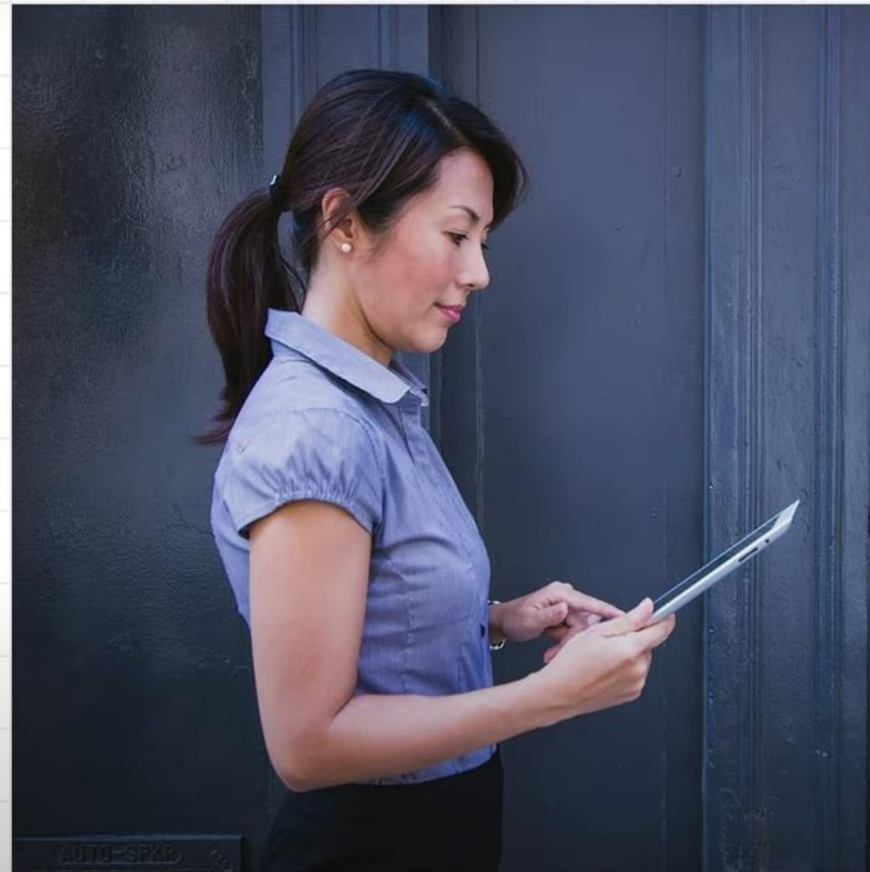




# Augmenting Images

```
# Updated to do image augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```





# Augmenting Images

```
# Updated to do image augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```









# Augmenting Images

```
# Updated to do image augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```

## Apa itu Dropout

**Dropout** merupakan proses mencegah terjadinya overfitting dan juga mempercepat proses learning. Dropout mengacu kepada menghilangkan neuron yang berupa hidden maupun layer yang visible di dalam jaringan. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada.

**Mengapa ini bisa membantu mengurangi overfitting?**

```
tf.keras.layers.Dropout(0.5)
```

# Effect on Augmentation

Cats and Dogs Dataset

<https://goo.gle/2AKklzX>



# QnA

**Thank you**