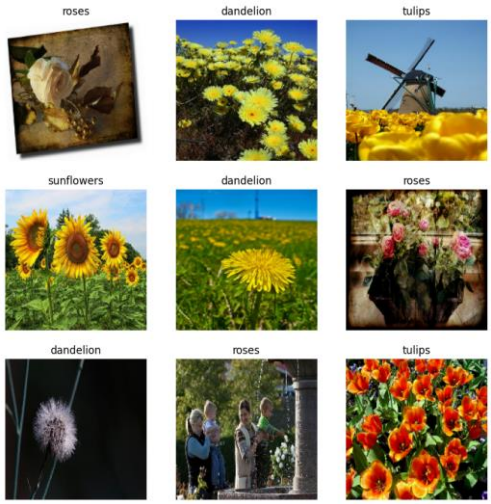
Logistic Classification Report

Data set:

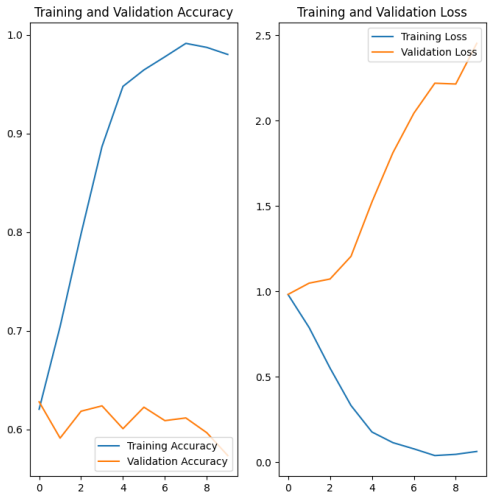
Google APIs data，there are 3670 photos belong to 5 classes.



Training model

```
model = Sequential([
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

Training results:

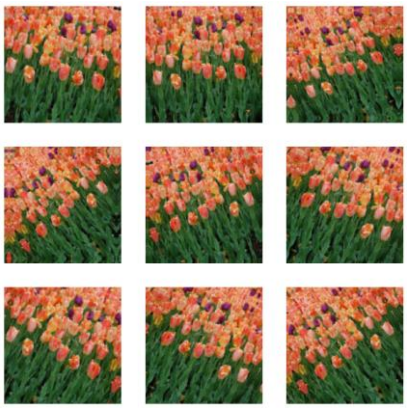


In the plots above, the training accuracy is increasing over time, whereas

validation accuracy stalls around 60% in the processing. It might be overfitting.
When there are a small number of training examples, the model sometimes learns form noises or unwanted details from training examples.
I use Data augmentation and dropout to fix this problem.

Data augmentation:

```
data_augmentation = keras.Sequential(
    [
        layers.RandomFlip('horizontal', input_shape=(img_height, img_width, 3)),
        layers.RandomRotation(0.1),
        layers.RandomZoom(0.1),
    ]
)
```

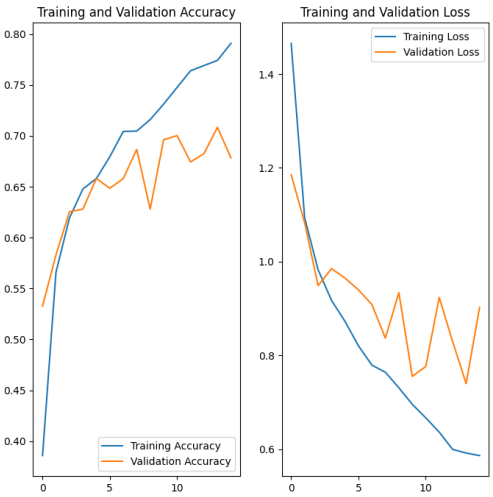


Data augmentation takes the approach of generating additional training data from your existing examples by augmenting them using random transformations that yield believable-looking images. This helps expose the model to more aspects of the data and generalize better.

New training model:

```
model = Sequential([
  data_augmentation,
  layers.Rescaling(1./255),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes, name="outputs")
])
```

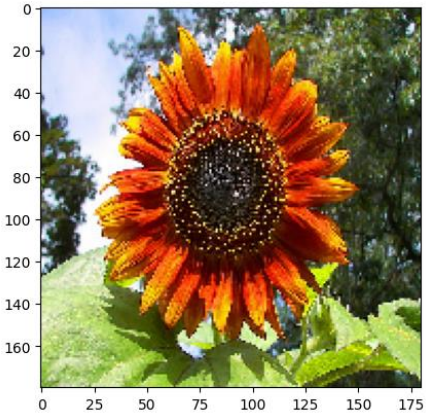
New training results:



After dropout and augmentation, result is better.

Prediction

Test data:



This image most likely to sunflowers with a 95.84 percent confidence. And it also showed that logistic classification can work well in the presences of moderate non- linearities.