LAPORAN AKHIR PRAKTIKUM

Mata Praktikum	: Kecerdasan Artifisial
Kelas	: 3IA19
Praktikum ke	: 2
Tanggal	: 10/26/2023
Materi	: Pengenalan Kecerdasan Artifisial
NPM	: 50421859
Nama	: Muhamad Ariel Dwi Prayoga
Asisten	: Muhammad Hauzan Dini Fakhri
Paraf Asisten	: -
Nama Asisten	: -
Jumlah Lembar	: 6

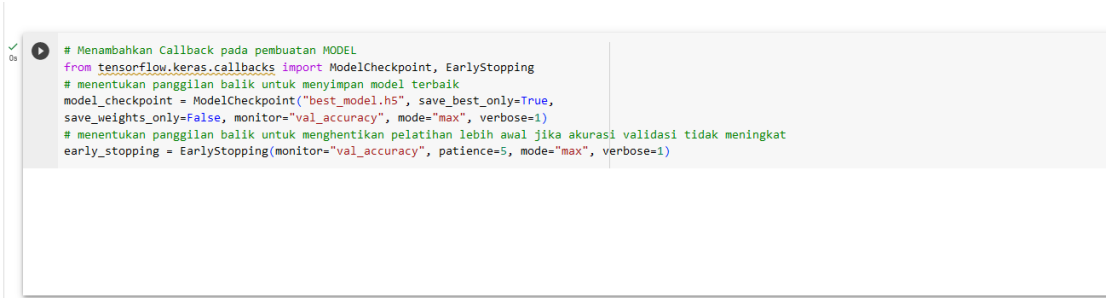
LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023

LISTING

1. Tambahkan Function Callback pada pembuatan Model Pada pertemuan 2 ini?



```
# Menambahkan Callback pada pembuatan MODEL
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
# menentukan panggilan balik untuk menyimpan model terbaik
model_checkpoint = ModelCheckpoint("best_model.h5", save_best_only=True,
save_weights_only=False, monitor="val_accuracy", mode="max", verbose=1)
# menentukan panggilan balik untuk menghentikan pelatihan lebih awal jika akurasi validasi tidak meningkat
early_stopping = EarlyStopping(monitor="val_accuracy", patience=5, mode="max", verbose=1)
```

Menambahkan Callback pada pembuatan MODEL

from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

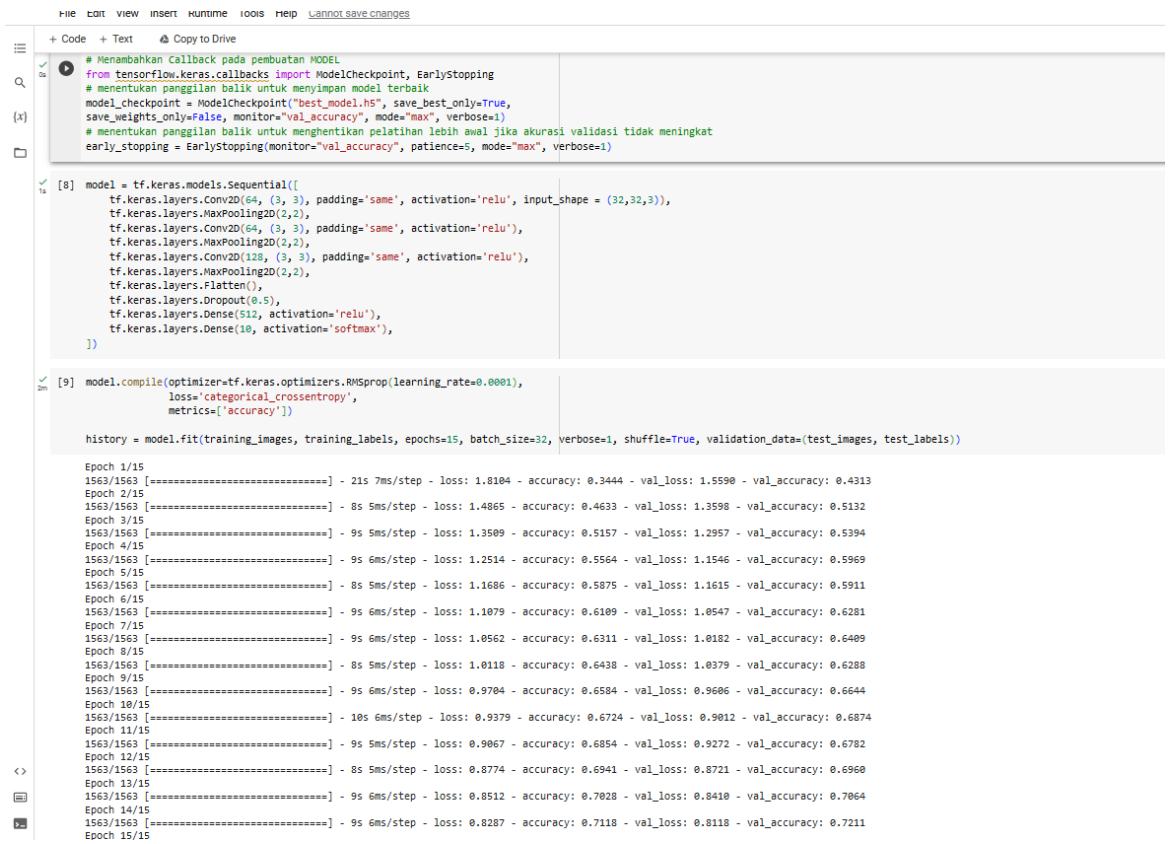
menentukan panggilan balik untuk menyimpan model terbaik

model_checkpoint = ModelCheckpoint("best_model.h5", save_best_only=True,

save_weights_only=False, monitor="val_accuracy", mode="max", verbose=1)

menentukan panggilan balik untuk menghentikan pelatihan lebih awal jika akurasi validasi tidak Meningkat

early_stopping = EarlyStopping(monitor="val_accuracy", patience=5, mode="max", verbose=1)



```
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive

# Menambahkan Callback pada pembuatan MODEL
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
# menentukan panggilan balik untuk menyimpan model terbaik
model_checkpoint = ModelCheckpoint("best_model.h5", save_best_only=True,
save_weights_only=False, monitor="val_accuracy", mode="max", verbose=1)
# menentukan panggilan balik untuk menghentikan pelatihan lebih awal jika akurasi validasi tidak meningkat
early_stopping = EarlyStopping(monitor="val_accuracy", patience=5, mode="max", verbose=1)

[8] model = tf.keras.models.Sequential([
        tf.keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu', input_shape = (32,32,3)),
        tf.keras.layers.MaxPooling2D(2,2),
        tf.keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu'),
        tf.keras.layers.MaxPooling2D(2,2),
        tf.keras.layers.Conv2D(128, (3, 3), padding='same', activation='relu'),
        tf.keras.layers.MaxPooling2D(2,2),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dense(10, activation='softmax'),
    ])

[9] model.compile(optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.0001),
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

    history = model.fit(training_images, training_labels, epochs=15, batch_size=32, verbose=1, shuffle=True, validation_data=(test_images, test_labels))

    Epoch 1/15
    1563/1563 [==============================] - 21s 7ms/step - loss: 1.8104 - accuracy: 0.3444 - val_loss: 1.5590 - val_accuracy: 0.4313
    Epoch 2/15
    1563/1563 [==============================] - 8s 5ms/step - loss: 1.4865 - accuracy: 0.4633 - val_loss: 1.3598 - val_accuracy: 0.5132
    Epoch 3/15
    1563/1563 [==============================] - 9s 5ms/step - loss: 1.3509 - accuracy: 0.5157 - val_loss: 1.2957 - val_accuracy: 0.5394
    Epoch 4/15
    1563/1563 [==============================] - 9s 6ms/step - loss: 1.2514 - accuracy: 0.5564 - val_loss: 1.1546 - val_accuracy: 0.5969
    Epoch 5/15
    1563/1563 [==============================] - 8s 5ms/step - loss: 1.1686 - accuracy: 0.5875 - val_loss: 1.1615 - val_accuracy: 0.5911
    Epoch 6/15
    1563/1563 [==============================] - 9s 6ms/step - loss: 1.1079 - accuracy: 0.6109 - val_loss: 1.0547 - val_accuracy: 0.6281
    Epoch 7/15
    1563/1563 [==============================] - 9s 6ms/step - loss: 1.0562 - accuracy: 0.6311 - val_loss: 1.0182 - val_accuracy: 0.6409
    Epoch 8/15
    1563/1563 [==============================] - 8s 5ms/step - loss: 1.0118 - accuracy: 0.6438 - val_loss: 1.0379 - val_accuracy: 0.6288
    Epoch 9/15
    1563/1563 [==============================] - 9s 6ms/step - loss: 0.9704 - accuracy: 0.6584 - val_loss: 0.9606 - val_accuracy: 0.6644
    Epoch 10/15
    1563/1563 [==============================] - 10s 6ms/step - loss: 0.9379 - accuracy: 0.6724 - val_loss: 0.9012 - val_accuracy: 0.6874
    Epoch 11/15
    1563/1563 [==============================] - 9s 5ms/step - loss: 0.9067 - accuracy: 0.6854 - val_loss: 0.9272 - val_accuracy: 0.6782
    Epoch 12/15
    1563/1563 [==============================] - 8s 5ms/step - loss: 0.8774 - accuracy: 0.6941 - val_loss: 0.8721 - val_accuracy: 0.6960
    Epoch 13/15
    1563/1563 [==============================] - 9s 6ms/step - loss: 0.8512 - accuracy: 0.7028 - val_loss: 0.8410 - val_accuracy: 0.7064
    Epoch 14/15
    1563/1563 [==============================] - 9s 6ms/step - loss: 0.8287 - accuracy: 0.7118 - val_loss: 0.8118 - val_accuracy: 0.7211
    Epoch 15/15
```

2. Kenapa kita menggunakan dataset Citar10. Berikan ulasan anda?

   Jawab:

   Dataset CIFAR-10 sering digunakan dalam penelitian dan pengembangan algoritma pembelajaran mesin dan penglihatan komputer karena beberapa alasan:

   1. **Ukuran Gambar**: Gambar dalam dataset ini memiliki dimensi 32x32, yang berarti mereka cukup kecil untuk pelatihan algoritma menjadi cepat, tetapi cukup besar untuk menjadi tantangan dan memungkinkan model untuk mempelajari fitur yang berguna.

   2. **Varietas Kelas**: Dataset ini terdiri dari 60.000 gambar berwarna dalam 10 kelas, dengan 6.000 gambar per kelas. Varietas ini memungkinkan model untuk belajar bagaimana membedakan antara berbagai objek.

   3. **Pelatihan dan Pengujian**: Dataset ini dibagi menjadi 50.000 gambar pelatihan dan 10.000 gambar pengujian. Ini memungkinkan peneliti untuk melatih model mereka pada satu set data dan kemudian menguji seberapa baik model tersebut generalisasi ke data yang belum pernah dilihat sebelumnya.

   4. **Benchmarking**: CIFAR-10 adalah standar industri untuk benchmarking algoritma pembelajaran mesin. Hasil pada dataset ini sering dilaporkan dalam literatur penelitian dan digunakan untuk membandingkan kinerja relatif dari algoritma yang berbeda[3].

   5. **Ketersediaan**: Dataset ini tersedia secara bebas untuk digunakan oleh komunitas penelitian, membuatnya menjadi pilihan yang populer.

   Secara keseluruhan, penggunaan dataset CIFAR-10 dapat membantu dalam pengembangan dan peningkatan algoritma pembelajaran mesin dan penglihatan komputer.

3. Kenapa kita membutuhkan normalisasi nilai piksel pada tahap data preprocessing. Dan kenapa harus dibagi 255?

   Normalisasi nilai piksel dalam pengolahan gambar adalah proses penting karena beberapa alasan:

   1. **Rentang Nilai**: Nilai piksel dalam gambar berwarna biasanya berkisar antara 0 hingga 255. Normalisasi dengan membagi semua nilai piksel dengan 255 mengubah rentang ini menjadi 0 hingga 1. Ini adalah praktik umum dalam pembelajaran mesin dan penglihatan komputer karena banyak algoritma bekerja lebih baik ketika mereka beroperasi pada angka yang lebih kecil.

   2. **Peningkatan Kinerja**: Dengan normalisasi, angka-angka menjadi lebih kecil dan perhitungan menjadi lebih mudah dan lebih cepat.

   3. **Pembelajaran Algoritma**: Banyak algoritma pembelajaran mesin bekerja lebih baik ketika data masukan memiliki rata-rata nol dan standar deviasi satu. Dengan mengurangi rata-rata dan membagi dengan standar deviasi, kita dapat mencapai ini.

   4. **Konsistensi Data**: Normalisasi memastikan bahwa semua data masukan ke model memiliki skala yang sama. Tanpa normalisasi, perbedaan skala antara fitur yang berbeda dapat menyebabkan model untuk belajar secara tidak efisien.

   5. **Meningkatkan Akurasi**: Normalisasi dapat membantu meningkatkan akurasi model pembelajaran mesin dan mengurangi waktu pelatihan.

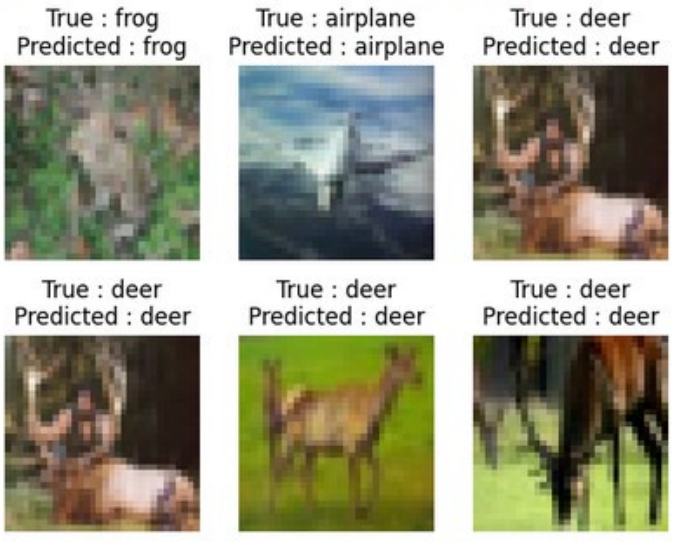
   Jadi, membagi nilai piksel dengan 255 adalah cara untuk mencapai normalisasi ini.

4. Jelaskan kodingan predikat model serta ubah menjadi ukuran plot menjdai lebih dari 2*2?

```
[12] dict_label = {0:'airplane', 1:'automobile', 2:'bird',
                   3:'cat', 4:'deer', 5:'dog',
                   6:'frog', 7:'horse', 8:'ship', 9:'truck'}

     def predict():
       some_random_number = np.random.randint(low=0, high=len(test_images)-1, size=6)
       sample_images = training_images[some_random_number]
       sample_label = training_labels[some_random_number]
       predicted = model.predict(sample_images)
       predicted = np.argmax(predicted, axis = 1)
       true_label = np.argmax(sample_label, axis = 1)
       fig, axs = plt.subplots(2,3)
       for row in range(2):
         for col in range(3):
           if row == 0:
             true = true_label[row+col]
             pred = predicted[row+col]
             axs[row, col].imshow(sample_images[row+col])
           else:
             true = true_label[row+col+1]
             pred = predicted[row+col+1]
             axs[row,col].imshow(sample_images[row+col+1])
           axs[row,col].set_title('True : %s\nPredicted : %s' % (dict_label[true], dict_label[pred]))
           axs[row,col].axis('off')

     predict()
```

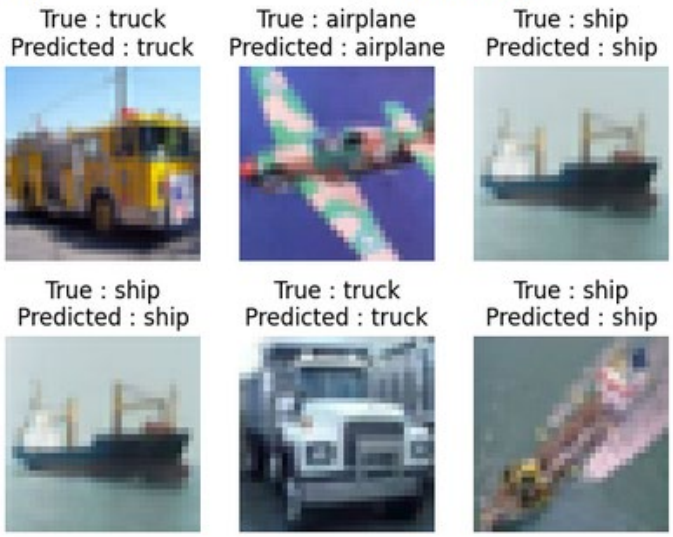
True

```
predict()

1/1 [==============================] - 0s 19ms/step
```



```
predict()
```

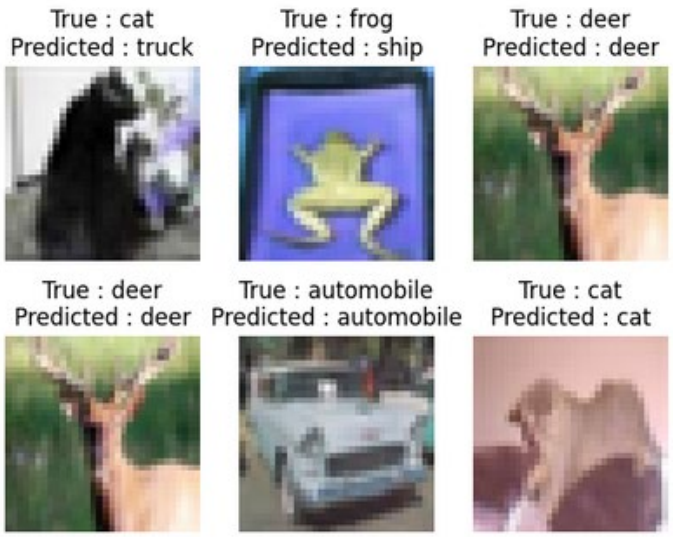
```
1/1 [==============================] - 0s 18ms/step
```



False

```
predict()
```

```
1/1 [==============================] - 0s 17ms/step
```



```
1/1 [==============================] - 0s 23ms/step
```