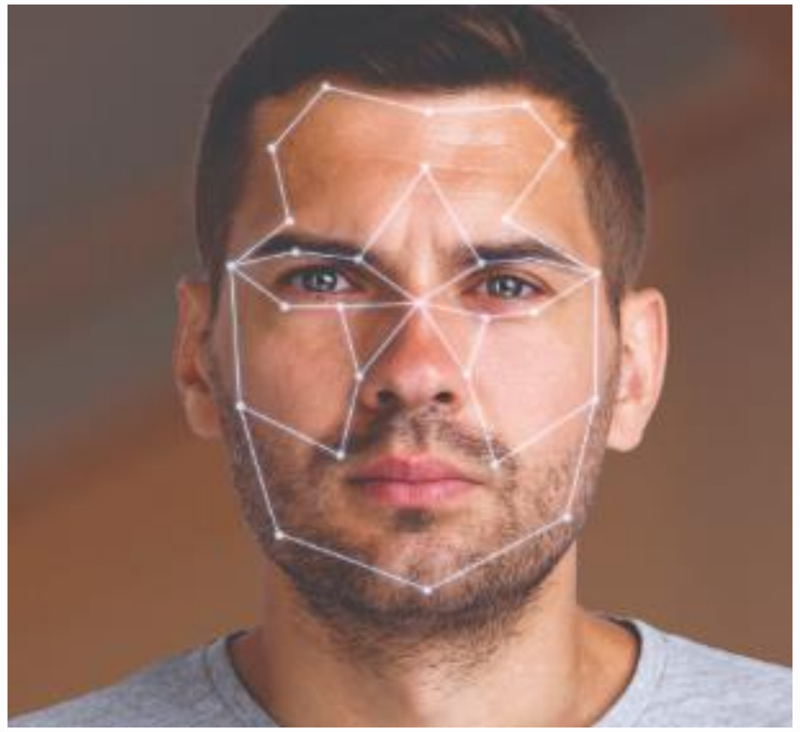
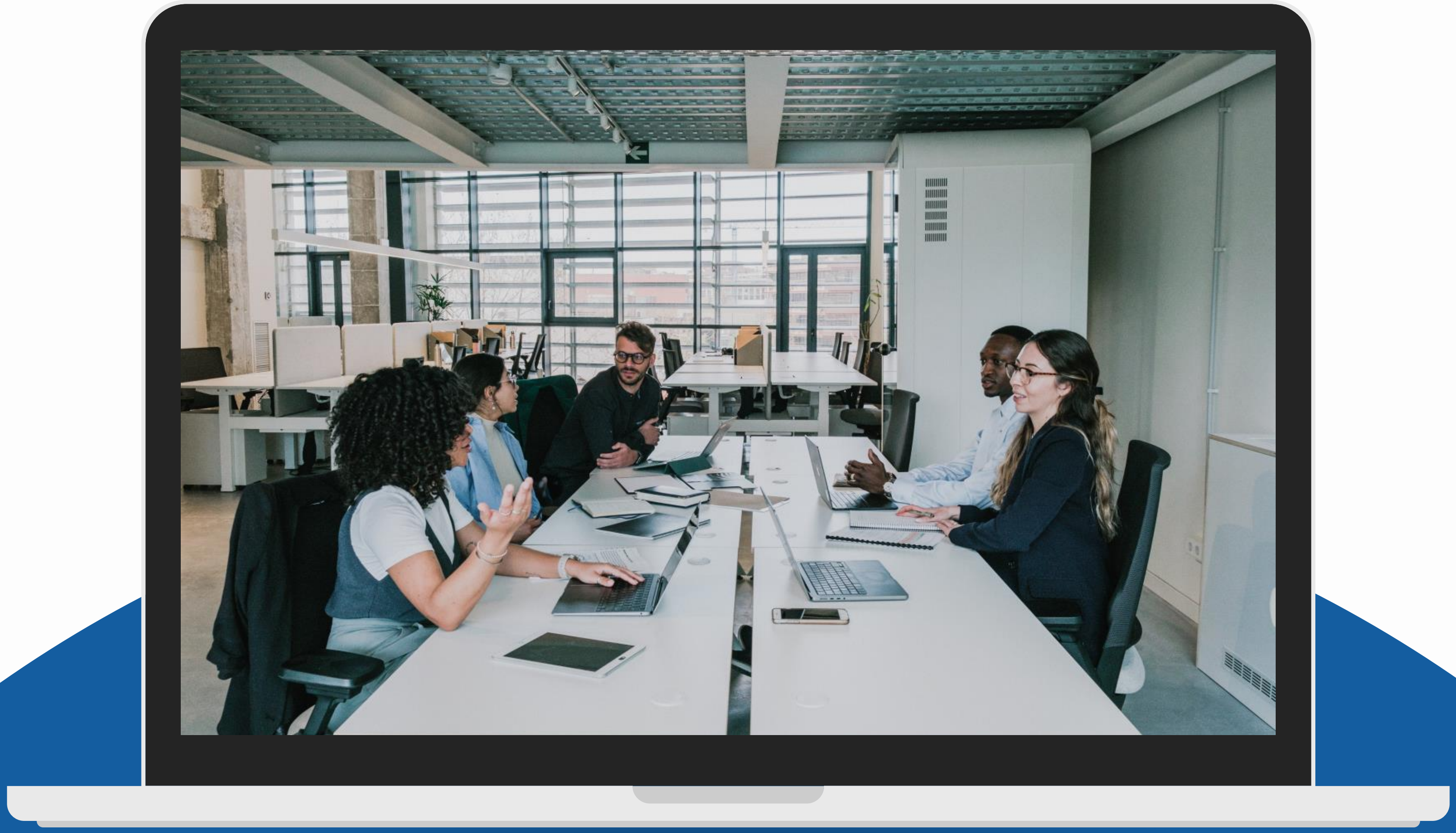
Portofolio Project 1
Face Recognition
(21 June 2024)

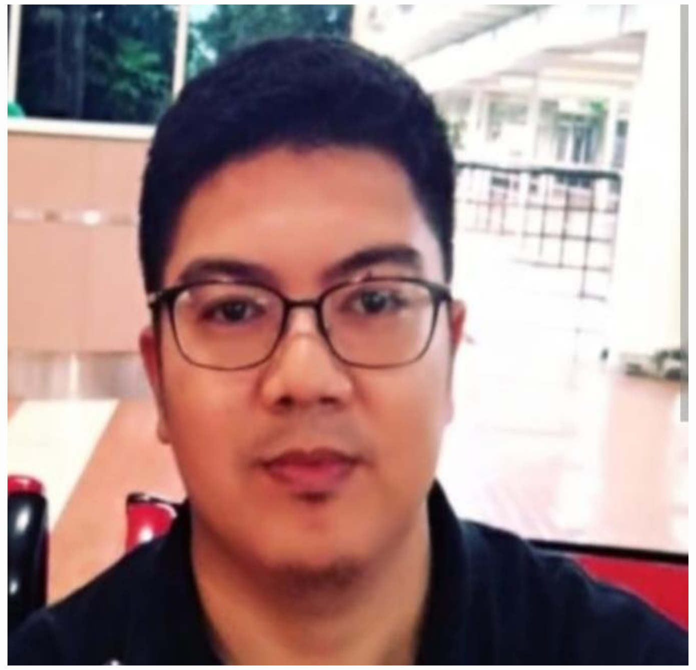




Kelompok CVA
Fei Fei Li

AI Career Bootcamp Batch 5
Computer Vison

Our Team



Rangga Etyawan
G.
AI Project Leader

Cahyadi Hartanto
S.
IT Expert

Yosabad Torando
Shalit
AI Engineer

Ade Rahman
AI Engineer

Machine Learning Life Cycle

Rangga Etyawan	Cahyadi Hartanto	Yosabad Torando	Ade Rahman
Business & Data Understanding	Data Cleaning & Analysis	Data Modelling	Business & Data Understanding
Model Evaluation	Data Modelling	Model Deployment	Data Cleaning & Analysis
PPT Preparation	Review Code	Review Code	Model Evaluation

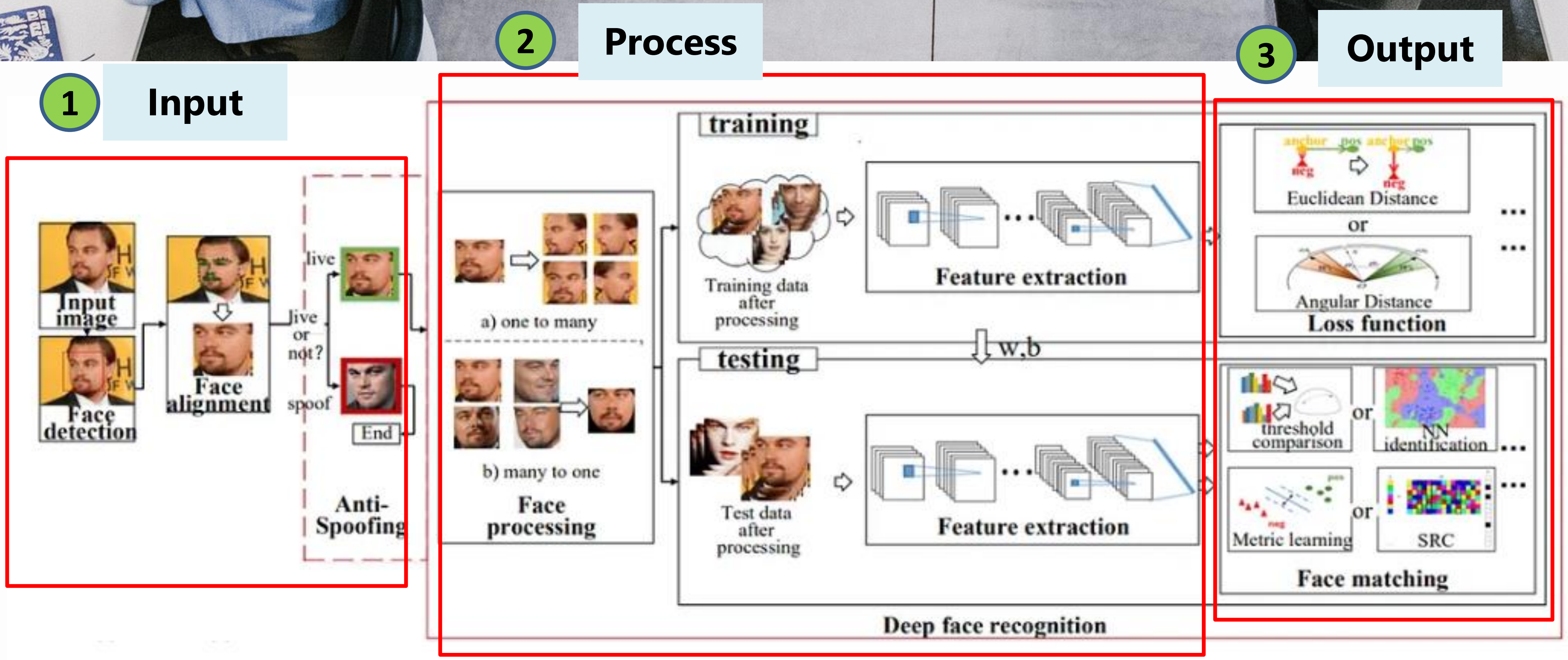
Introduction-Face Recognition

Face Recognition adalah Perangkat Lunak Analisis Kecerdasan Buatan Vision

yang menyediakan pencarian dan pengenalan akurat dengan mekanisme pencarian 1:N.

Menggunakan Algoritma Deep Learning untuk meningkatkan akurasi inferensi wajah.

Face Recognition secara otomatis melakukan Verifikasi Wajah dan Identifikasi Wajah yang mendeteksi fitur wajah dan mengklasifikasikannya menjadi orang yang dikenali atau tidak..



Face Recognition with Deep Learning

Face Recognition Model

Paper Research Face Recognition

VGG-16

Face Recognition Algorithm Based on VGG Network Model and SVM

Hongling Chen[1,*] and Chen Haoyu[2]

[1] College of computer and information science, Southwest University, China
[2] College of mathematics and Statistics, Southwest University, China

[*]Email: chenhongling@email.swu.edu.cn

Abstract. The problem that the dimension of facial features is too large does exist with the Deep learning face recognition. This paper proposes a face recognition algorithm based on SVM combined with VGG network model extracting facial features, which can not only accurately extract face features, but also reduce feature dimensions and avoid irrelevant features to participate in the calculation. Firstly, the VGG-16 model is obtained by training the training data set, which is used for feature extraction, on top of this, principal component analysis method (PCA) is used for feature dimensionality reduction, and last, the face recognition is performed by SVM classifier with linear kernel function. In this paper, we conduct a comparative experiment on CelebA dataset and find that the accuracy reaches its peak when the feature dimension is reduced to 400.The experiment is carried out on LFW dataset using 400-dimensional feature data, and comparing with other algorithms, the results show that the algorithm in this paper has reached the level of state-of-art.

Dimension Reduced to 400

ResNet 152v2

Face Recognition Method Based on Residual Convolution Neural Network

Arshi Husain[1] and Virendra P. Vishvakarma[2]
USICT, Guru Gobind Singh Indraprastha University
arshihusaincdac@gmail.com, virendravishwa@rediffmail.com, vpv@ipu.ac.in

Abstract. With the advancement of information technology and societal growth, social security has become more important than ever. Face recognition, as compared to other traditional recognition methods like fingerprint recognition, palm recognition, etc, has the benefit that it is contact less, and now it is becoming one among the most prominent technologies in development. Although there are numerous recognition systems that use DNNs in the field of facial expression recognition, their accuracy and practicality are still insufficient for real-world applications. A facial recognition approach based on Resnet 152 v2 has been proposed in this work. In this paper, a residual learning approach is presented to make the training of networks that are far deeper than previously employed networks easier. The proposed method, employs the AT&T face dataset, and supposing that normalization and segmentation are complete, we concentrate on the subtask of person verification and recognition, demonstrating performance using a testing database comprising illumination, pose, expression and occlusion variations. SoftMax is the activation function that has been used, which adjusts the output sum up to one allowing it to be understood as probabilities. Then, the model would generate a judgment depending on which option has a strong likelihood. This system employs Adam as an optimizer to control the learning rate through training and categorical cross entropy as its loss function. The proposed approach has a 97 percent face recognition accuracy on AT&T dataset, showing its efficacy after a significant number of analyses and experimental verification.

Keywords: Deep convolutional neural network, Face recognition, ResNet.

Accuracy 97%

GoogleNet

Research Article

Research on Face Recognition Classification Based on Improved GoogleNet

Zhigang Yu, Yunyun Dong, Jihong Cheng, Miaomiao Sun, and Feng Su

Yantai Nanshan University, Yantai 265713, China

Correspondence should be addressed to Miaomiao Sun; yuzhigang@nanshan.edu.cn

Received 19 November 2021; Revised 8 December 2021; Accepted 13 December 2021; Published 5 January 2022

Academic Editor: Jian Su

Face recognition is a relatively mature technology, which has some applications in many aspects, and now there are many networks studying it, which has indeed brought a lot of convenience to mankind in all aspects. This paper proposes a new face recognition technology. First, a new GoogLeNet-M network is proposed, which improves network performance on the basis of streamlining the network. Secondly, regularization and migration learning methods are added to improve accuracy. The experimental results show that the GoogLeNet-M network with regularization using migration learning technology has the best performance, with a recall rate of 0.97 and an accuracy of 0.98. Finally, it is concluded that the performance of the GoogLeNet-M network is better than other networks on the dataset, and the migration learning method and regularization help to improve the network performance.

Accuracy 98%

Schedule Project

1. Timeline Project

Group Discussion (7 June) → Data & Algoritma Understanding (8-13 June) → Group Discussion (14 June) → lodel Training & Evaluatio (15-20 June) → Project Presentation (21 June)

2. General Schedule

General Schedule Project 1 AI Career Computer Vision Bootcamp Batch 5 (Face Recognition)

No	Activity	Purpose	Jun-24								
			13-Jun-24	14-Jun-24	15-Jun-24	16-Jun-24	17-Jun-24	18-Jun-24	19-Jun-24	20-Jun-24	21-Jun-24
1	Group Discussion With Mentor (Kak Galang)	Data & Algoritma Understanding									
2	Group Discussion CVA Fei Fei Li	1. Review Feedback From Mentor 2. Confirm Jobdesk Team Member 3. Dataset Analysis 4. Model VGG Analysis									
3	Group Discussion CVA Fei Fei Li	1. Review Progress PPT & Code 2. Model GoogleNet & ResNet Analysis 3. Model Evaluation									
4	AI Project Evaluation (Schedule Dashboard Indonesia AI)	1. Review Progress PPT **(must finish)** & Code 2. Model VGG,GoogleNet & ResNet Analysis 3. Model Evaluation 4. Compare Optimal Model for Face Recognition									
5	Project Presentation	1. Presentation Project 1 (Face Recognition)									

Start Group Discussion

Project Presentation

Flowchart Project

Start

Dataset & Algoritma Understanding

Analisa Data Tabular & Gambar *List_attribute.csv & image (.jpg)

Penyesuaian Data Tabular & Gambar

Female	118.165	58%
Male	84.434	42%
	202.599	

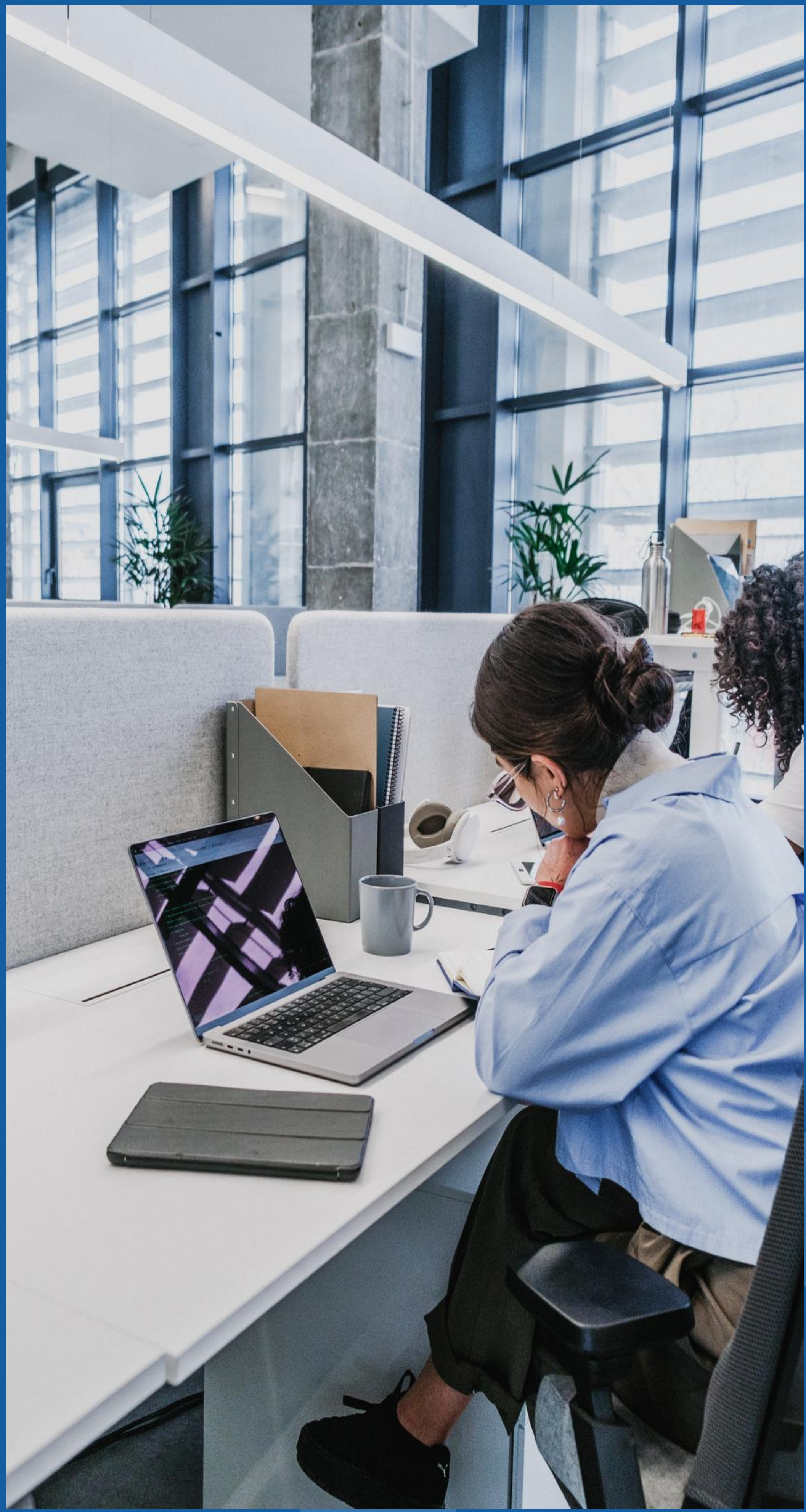
Pemrosesan Data Gambar

Female	2.953	59%
Male	2.047	41%
	5.000	

Modelling *Framework : TensorFlow →5 Model (VGG-16 & 19, ResNet-50 & 101, GoogleNet-V3)

Evaluation *Fix Parameter Batch Size : 32, Metric : Accuracy

End

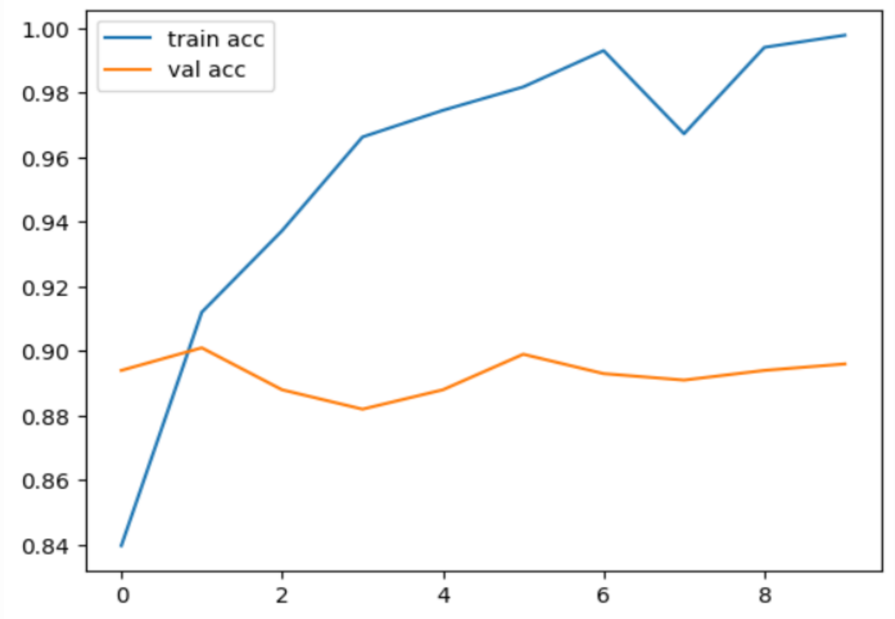


VGG Model

VGG-16 (Environmet:CPU)

```
# Modelling menggunakan VGG-16

vgg16 = VGG16(input_shape=(128, 128, 3),
weights='imagenet', include_top=False)
for layer in vgg16.layers:
  layer.trainable = False

x = vgg16.output
x = Flatten()(x)
x = Dense(1024, activation='relu')(x)
prediction = Dense(1, activation='sigmoid')(x)

model_vgg16 = Model(inputs=vgg16.input,
outputs=prediction)
model_vgg16.summary()
```

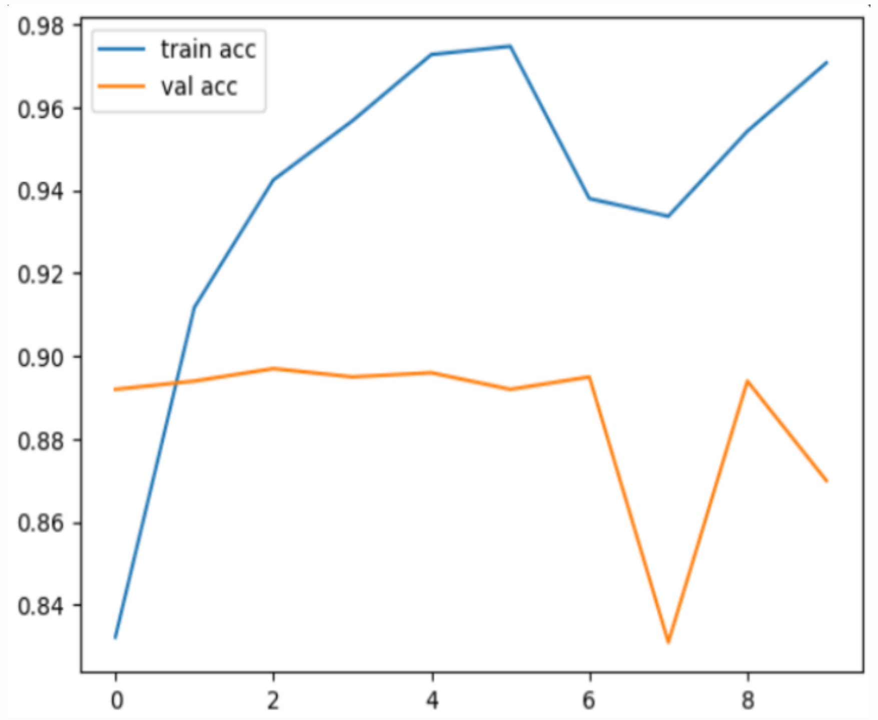
Epoch	Accuracy	Val.Accuracy	Train Time (s)	Train Speed (ms/step)
1	0,7526	0,8940	256	2
2	0,9208	0,9010	257	2
3	0,9367	0,8880	240	2
4	0,9660	0,8820	281	2
5	0,9755	0,8880	228	2
6	0,9884	0,8990	288	2
7	0,9553	0,8930	240	2
8	0,9805	0,8910	297	2
9	0,9932	0,8940	233	2
10	0,9975	0,8960	287	2
Max	0,9975	0,9010	297	2
Min	0,7526	0,8820	228	2
Average	0,9467	0,8926	261	2



VGG-19 (Environmet:CPU)

```
# Modelling menggunakan VGG-19

vgg = VGG19(input_shape=(128, 128, 3), weights='imagenet',
include_top=False)
for layer in vgg.layers:
  layer.trainable = False

x = vgg.output
x = Flatten()(x)
x = Dense(1024, activation='relu')(x)
prediction = Dense(1, activation='sigmoid')(x)

model_vgg19 = Model(inputs=vgg.input, outputs=prediction)
model_vgg19.summary()
```

Epoch	Accuracy	Val.Accuracy	Train Time (s)	Train Speed (ms/step)
1	0,7370	0,8920	378	3
2	0,9029	0,8940	383	3
3	0.9398	0,8970	356	3
4	0,9559	0,8950	359	3
5	0,9732	0,8960	354	3
6	0,9776	0,8920	355	3
7	0,9563	0,8950	361	3
8	0,9454	0,8310	356	3
9	0,9419	0,8940	373	3
10	0,9705	0,8700	405	3
Max	0,9776	0,8970	405	3
Min	0,7370	0,8310	354	3
Average	0,9290	0,8856	368	3



Evaluasi Model VGG

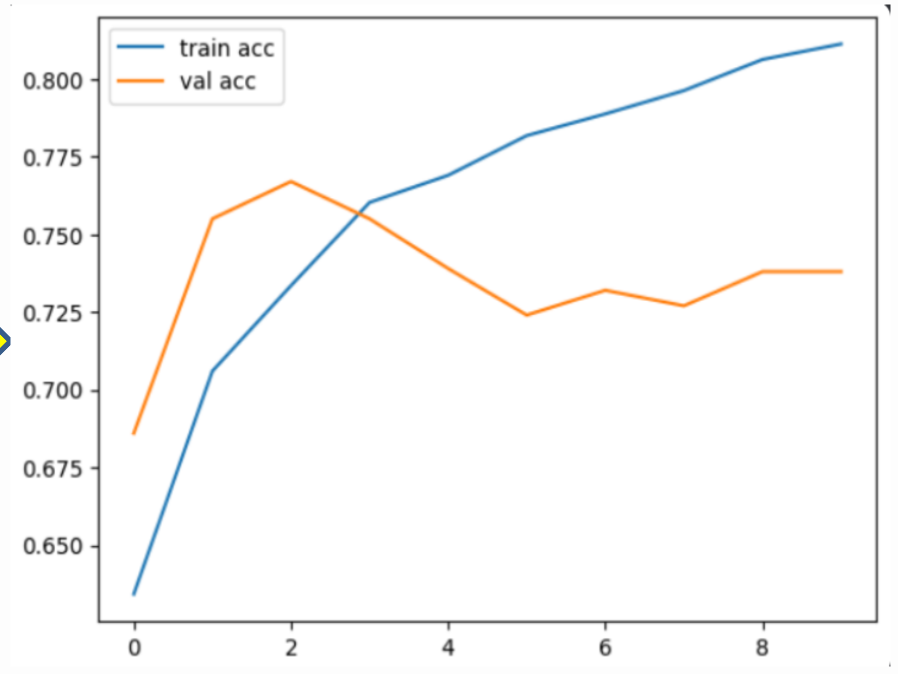
1. VGG-16 pada Epoch-10 memiliki Accuracy 0,9975, Validasi Accuracy 0,8960 dengan Train Time 287 s
2. VGG-19 pada Epoch-10 memiliki Accuracy 0,9705, Validasi Accuracy 0,8700 dengan Train Time 405 s
3. Evaluasi Model VGG-16 memiliki performa lebih baik daripada VGG-19 dengan Average Valiadasi Accuracy 0,9467

ResNet Model

ResNet-50 (Environment : GPU)

```
# Modelling menggunakan ResNet-50
from tensorflow.keras.applications import ResNet50

res = ResNet50(input_shape=(128, 128, 3), weights='imagenet'
include_top=False)
for layer in res.layers:
  layer.trainable = False

x = res.output
x = Flatten()(x)
x = Dense(1024, activation='relu')(x)
prediction = Dense(1, activation='sigmoid')(x)

model_res = Model(inputs=res.input, outputs=prediction)
model_res.summary()
```

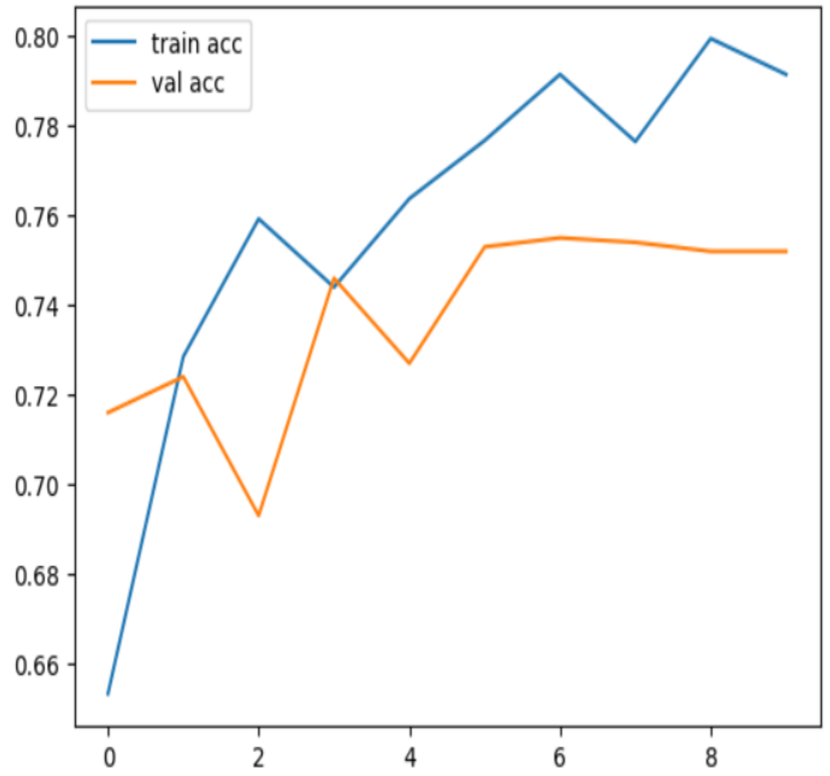
Epoch	Accuracy	Val.Accuracy	Train Time (s)	Train Speed (ms/step)
1	0,5196	0,6860	191	1
2	0,6723	0,7570	164	1
3	0,7076	0,7670	181	1
4	0,7374	0,7550	184	1
5	0,7440	0,7390	178	1
6	0,7555	0,7240	186	1
7	0,7655	0,7320	163	1
8	0,7722	0,7270	148	1
9	0,7879	0,7380	146	1
10	0,7949	0,7380	143	1
Max	0,7949	0,7670	191,0000	1,0000
Min	0,5196	0,6860	143,0000	1,0000
Average	0,7257	0,7363	168,4000	1,0000



ResNet-101 (Environment : CPU)

```
# Modelling menggunakan ResNet-101
res101 = ResNet101(input_shape=(128, 128, 3),
weights='imagenet', include_top=False)
for layer in res101.layers:
    layer.trainable = False

x = res101.output
x = Flatten()(x)
x = Dense(1024, activation='relu')(x)
prediction = Dense(1, activation='sigmoid')(x)

model_res101 = Model(inputs=res101.input, outputs=prediction)
model_res101.summary()
```

Epoch	Accuracy	Val.Accuracy	Train Time (s)	Train Speed (ms/step)
1	0,6504	0,7160	304	2
2	0,7204	0,7240	259	2
3	0,7691	0,6930	278	2
4	0,7364	0,7460	246	2
5	0,7755	0,7270	248	2
6	0,7740	0,7530	246	2
7	0,8019	0,7550	247	2
8	0,7804	0,7540	254	2
9	0,8020	0,7520	298	2
10	0,8036	0,7520	249	2
Max	0,8036	0,7550	304	2
Min	0,6504	0,6930	246	2
Average	0,7614	0,7372	263	2



Evaluasi Model ResNet

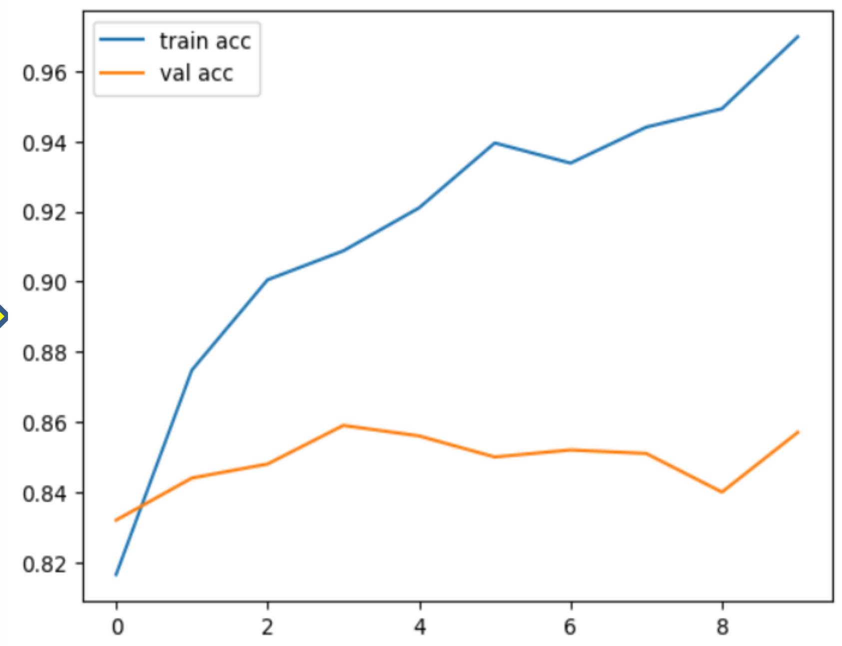
1. ResNet-50 pada Epoch-10 memiliki Accuracy 0,7949, Validasi Accuracy 0,7380 dengan Train Time 143 s
2. ResNet-101 pada Epoch-10 memiliki Accuracy 0,8306, Validasi Accuracy 0,7520 dengan Train Time 249 s
3. Evaluasi Model ResNet-101 memiliki performa lebih baik daripada ResNet-50 dengan Average Valiadasi Accuracy 0,7614

GoogleNet Model

GoogleNet-V3 (Environmet : CPU)

```
# Modelling menggunakan InceptionV3
incv3 = InceptionV3(input_shape=(128, 128, 3),
weights='imagenet', include_top=False)
for layer in incv3.layers:
    layer.trainable = False

x = incv3.output
x = Flatten()(x)
x = Dense(1024, activation='relu')(x)
prediction = Dense(1, activation='sigmoid')(x)

model_incv3 = Model(inputs=incv3.input, outputs=prediction)
model_incv3.summary()
```

Epoch	Accuracy	Val.Accuracy	Train Time (s)	Train Speed (ms/step)
1	0,7641	0,8320	64	437
2	0,8817	0,8440	66	525
3	0,9150	0,8480	61	488
4	0,9170	0,8590	60	478
5	0,9252	0,8560	72	577
6	0,9457	0,8500	87	698
7	0,9409	0,8520	67	537
8	0,9430	0,8510	60	478
9	0,9391	0,8400	59	471
10	0,9674	0,8570	59	474
Max	0,9674	0,8590	87	698
Min	0,7641	0,8320	59	437,0000
Average	0,9139	0,8489	66	516,3000



Evaluasi Model GoogleNet

1. GoogleNet-V3 pada Epoch-10 memiliki Accuracy 0,9674, Validasi Accuracy 0,8570 dengan Train Time 474 s
2. Evaluasi Model GoogleNet-V3 memiliki performa Average Valiadasi Accuracy 0,9139

Sampel Deployment

Parameter	VGG-16	VGG-19	ResNet-50	ResNet-101	GoogleNet-V3
1. Accuracy	0,9975	0,9705	0,7949	0,8306	0,9674
2. Val.Accuracy	0,8960	0,8700	0,7380	0,7520	0,8570

Evaluasi Validasi Tertinggi Model VGG-16

Source Code

```
# Load model yang telah dilatih
model = tf.keras.models.load_model('vgg16_model.h5')

# Path ke folder Test
test_folder_path = 'Test' #gambar-gambar masukin sini

import time

# List untuk menyimpan gambar dan nama file
test_images = []
test_filenames = []

def load_and_preprocess_test_image(file_name, target_size=
(128, 128)):
    image_path = os.path.join(test_folder_path, file_name)
    image = Image.open(image_path)
    image = image.resize(target_size)
    image = np.array(image) / 255.0  # Normalisasi gambar
    return image

# Loop untuk load dan preprocess gambar di folder Test
for filename in os.listdir(test_folder_path):
    image = load_and_preprocess_test_image(filename)
    test_images.append(image)
```

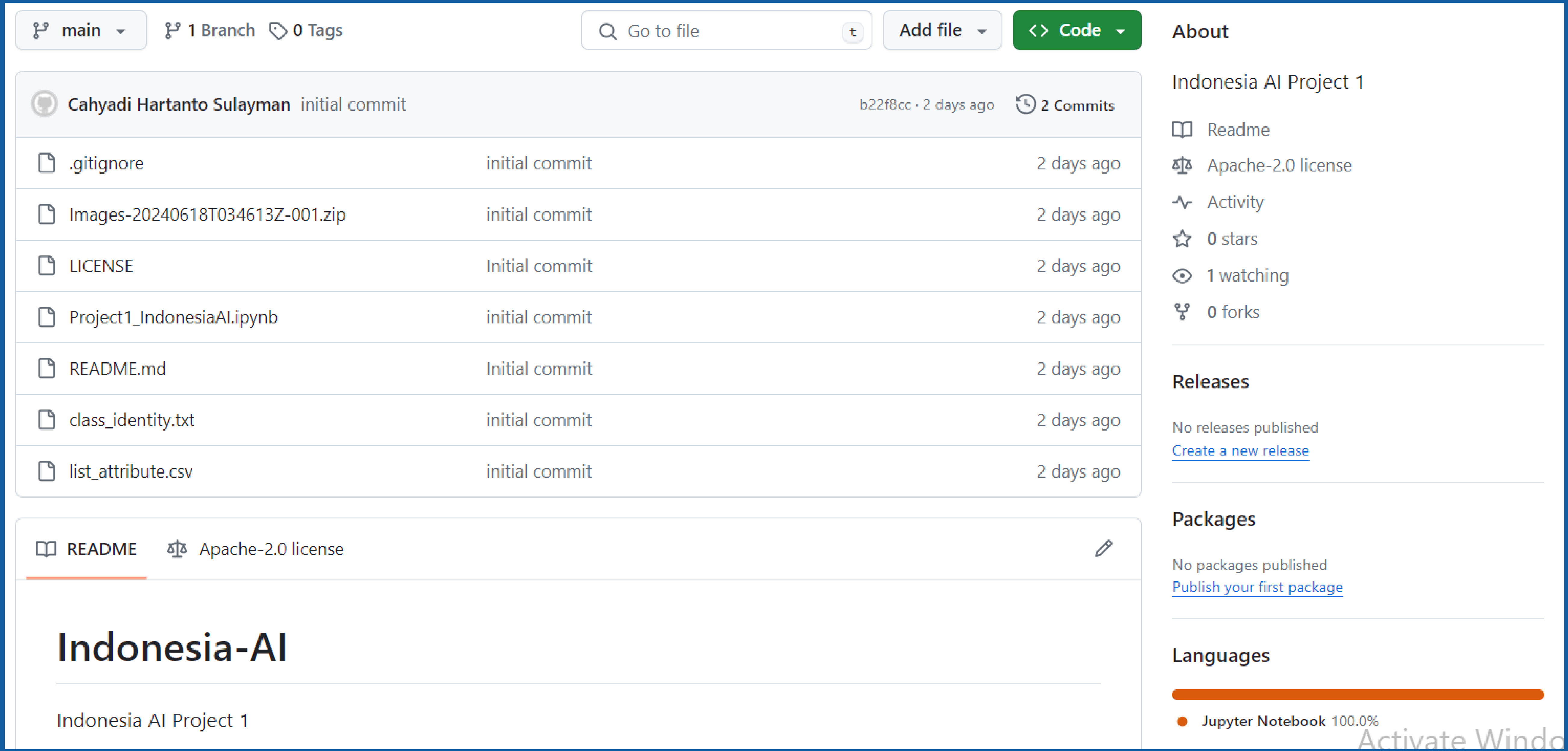
```
# Konversi ke numpy array
test_images = np.array(test_images)

# Prediksi
start_time = time.time()
predictions = model.predict(test_images)
end_time = time.time()

# Menampilkan hasil prediksi
for i in range(len(test_filenames)):
    gender = 'Male' if predictions[i] > 0.5 else 'Female'
    print(f"File: {test_filenames[i]}, Predicted Gender:
{gender}")

print()
print(f"Total prediction time: {end_time - start_time}
seconds")
```

Result Predicted Gender

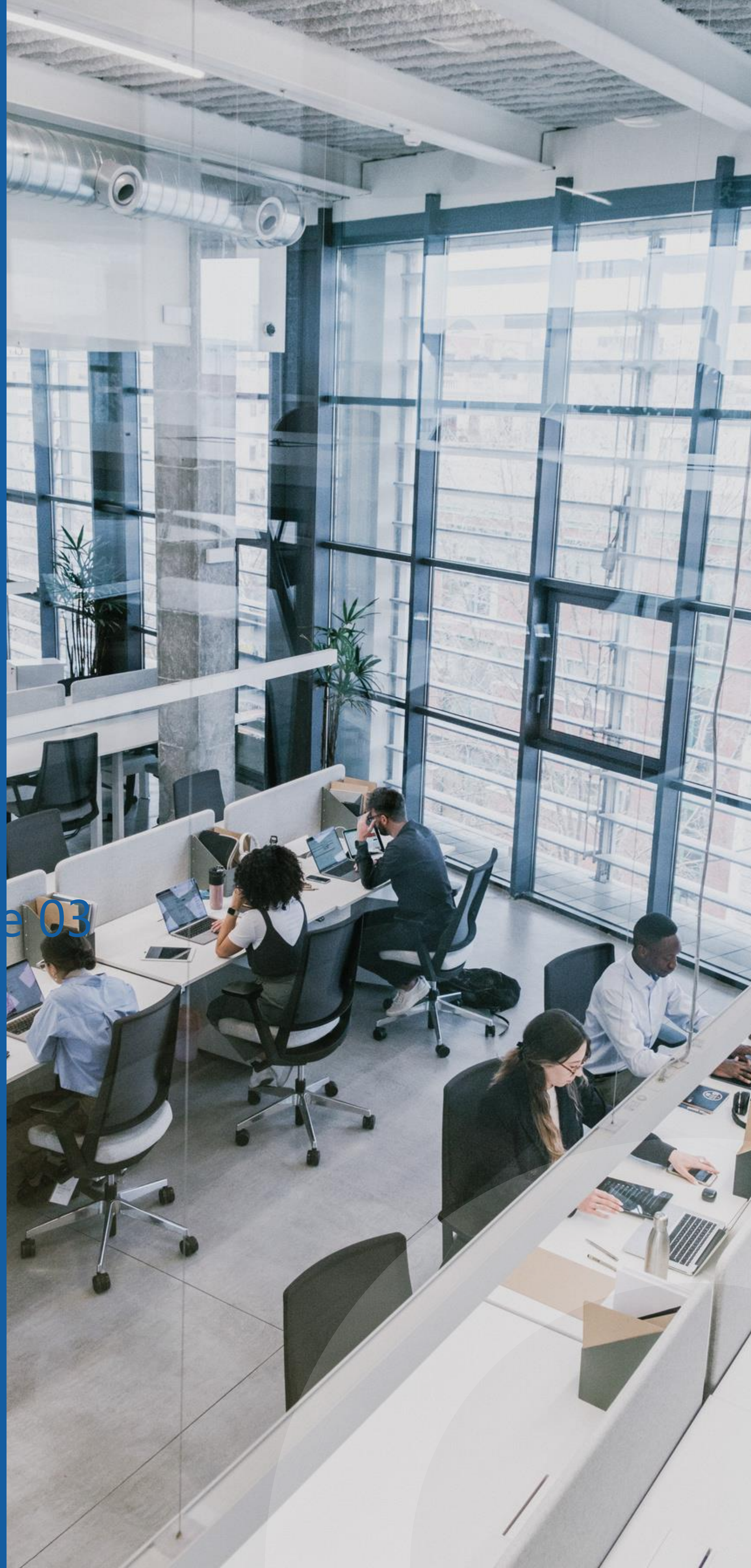
```
1/1 [==============================] - 0s 419ms/step
File: cew1.jpeg, Predicted Gender: Female
File: cew2.jpg, Predicted Gender: Female
File: cew3.jpeg, Predicted Gender: Female
File: cew4.jpg, Predicted Gender: Female
File: cew5.jpg, Predicted Gender: Female
File: cow1.jpg, Predicted Gender: Male
File: cow2.jpeg, Predicted Gender: Male
File: cow3.jpeg, Predicted Gender: Male
File: cow4.jpg, Predicted Gender: Male
File: cow5.jpeg, Predicted Gender: Male

Total prediction time: 0.4479053020477295 seconds
```

Inference Time : 0,04479053020477295 S

Github



main
1 Branch
0 Tags
Go to file
t
Add file
Code
Cahyadi Hartanto Sulayman initial commit
b22f8cc · 2 days ago
2 Commits
.gitignore initial commit 2 days ago
Images-20240618T034613Z-001.zip initial commit 2 days ago
LICENSE Initial commit 2 days ago
Project1_IndonesiaAI.ipynb initial commit 2 days ago
README.md Initial commit 2 days ago
class_identity.txt initial commit 2 days ago
list_attribute.csv initial commit 2 days ago
README
Apache-2.0 license
Indonesia-AI
Indonesia AI Project 1
About
Indonesia AI Project 1
Readme
Apache-2.0 license
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Jupyter Notebook 100.0%

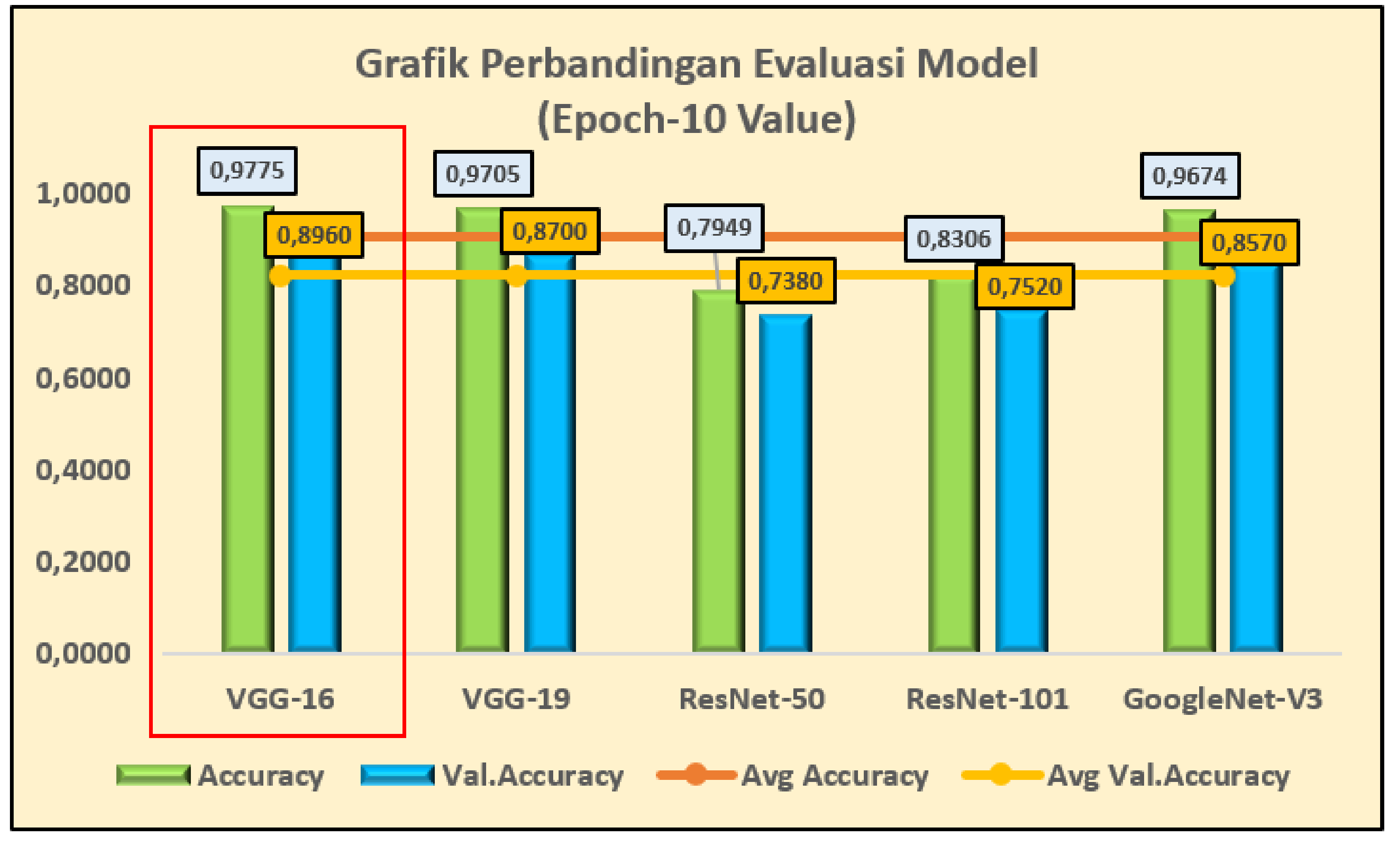


Kesimpulan

1. Hasil Evaluasi Model

Epoch-10 Value

Parameter	VGG-16	VGG-19	ResNet-50	ResNet-101	GoogleNet-V3	Max	Min	Average
1. Accuracy	0,9975	0,9705	0,7949	0,8306	0,9674	0,9775	0,7949	0,9082
2. Val.Accuracy	0,8960	0,8700	0,7380	0,7520	0,8570	0,8960	0,7380	0,8226



POV Hasil Evaluasi Model-Epoch-10 Value

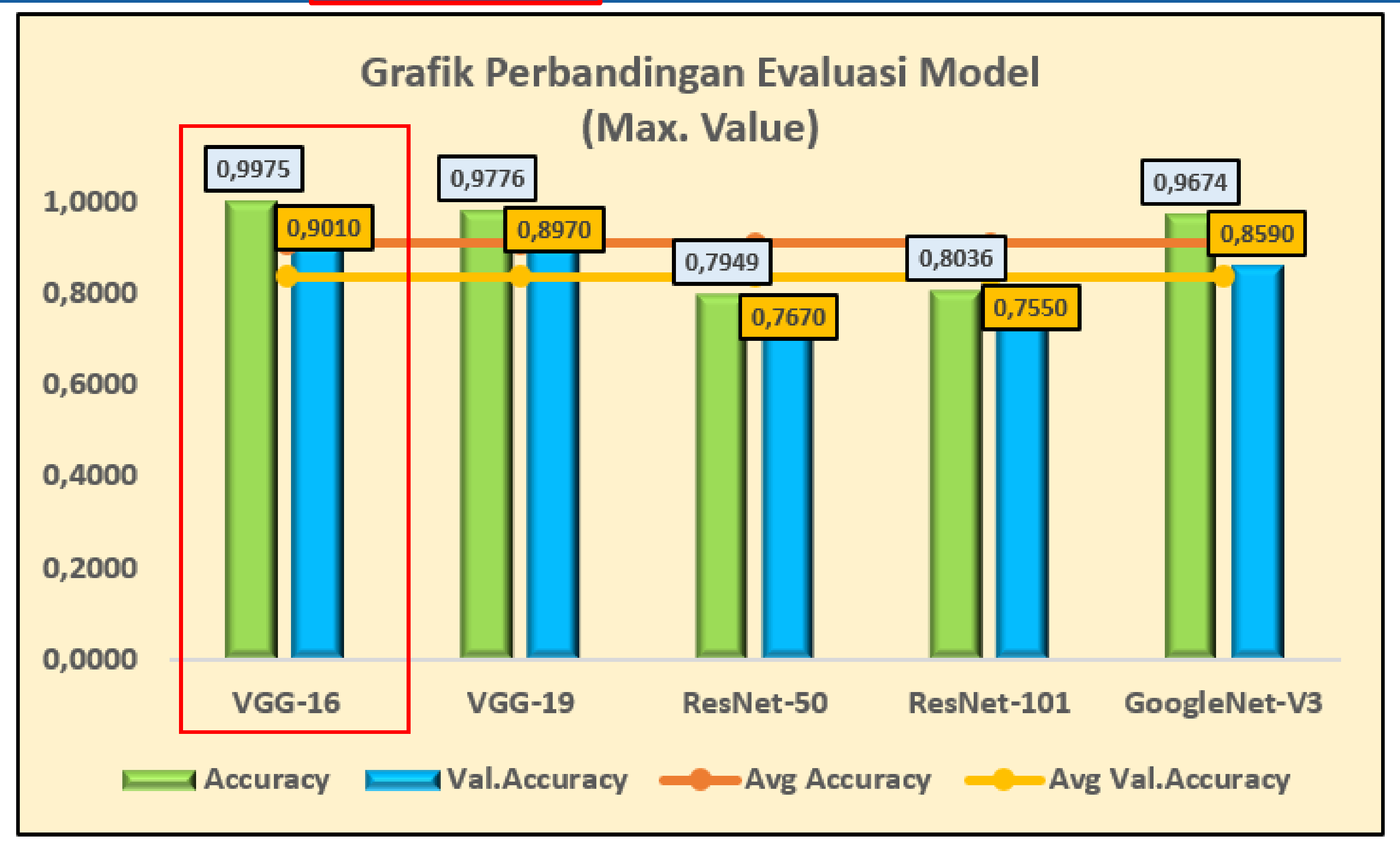
1. **Perbandingan Evaluasi Model pada Epoch-10 Value, VGG-16 menghasilkan Hasil Evaluasi Model paling optimal dengan Accuracy 0,9775 & Validasi Accuracy 0,8960**
2. **Inference Time**

Kesimpulan

2. Hasil Evaluasi Model

Max.Value

Parameter	VGG-16	VGG-19	ResNet-50	ResNet-101	GoogleNet-V3	Max	Min	Average
1. Accuracy	0,9975	0,9776	0,7949	0,8036	0,9674	0,9975	0,7949	0,9082
2. Val.Accuracy	0,9010	0,8970	0,7670	0,7550	0,8590	0,9010	0,7550	0,8358



POV Hasil Evaluasi Model Max.Value

1. **Perbandingan Evaluasi Model pada Max.Value, VGG-16 menghasilkan Hasil Evaluasi Model paling optimal dengan Accuracy 0,9975 & Validasi Accuracy 0,9010**

Kesimpulan

3. Hasil Evaluasi Model VGG-16 lebih baik dibanding Model Lain (ResNet-50 & ResNet-101)

Epoch-10 Value

Parameter	VGG-16	VGG-19	ResNet-50	ResNet-101	GoogleNet-V3	Max	Min	Average
1. Accuracy	0,9975	0,9705	0,7949	0,8306	0,9674	0,9775	0,7949	0,9082
2. Val.Accuracy	0,8960	0,8700	0,7380	0,7520	0,8570	0,8960	0,7380	0,8226

VGG-16 Memiliki Hasil Evaluasi Model Lebih Baik dibanding ResNet-50 & ResNet-101

1. **Pre-Trained Model VGG-16 memiliki kompleksitas layer yang lebih baik untuk diimplementasikan pada kasus klasifikasi wajah dibanding ResNet.**
2. **ResNet yang memiliki layer lebih dalam memerlukan lebih banyak epoch untuk mencapai konvergensi.**

Continous Improvement

1. Meningkatkan Akurasi ResNet-50 & ResNet-101 (Model dengan Hasil Evaluasi Accuracy Terendah)

Parameter	VGG-16	VGG-19	ResNet-50	ResNet-101	GoogleNet-V3	Max	Min	Average
1. Accuracy	0,9975	0,9705	0,7949	0,8306	0,9674	0,9775	0,7949	0,9082
2. Val.Accuracy	0,8960	0,8700	0,7380	0,7520	0,8570	0,8960	0,7380	0,8226

1. Melakukan Augmentasi Data Image

Source Code

```
# Fungsi untuk augmentasi
def train_val_generators(TRAINING_DIR, VALIDATION_DIR, size):
   train_datagen = ImageDataGenerator(
      rescale=1./255.,
      rotation_range=20,
      zoom_range=0.2,
      shear_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest'
   )
   train_generator = train_datagen.flow_from_directory(
      directory=TRAINING_DIR,
      batch_size=32,
      target_size=size,
      class_mode='binary'
   )

   validation_datagen = ImageDataGenerator(
      rescale=1./255.
   )
   validation_generator =
validation_datagen.flow_from_directory(
      directory=VALIDATION_DIR,
      batch_size=32,
      target_size=size,
      class_mode='binary'
   )
   return train_generator, validation_generator
```

```
train_generator_res, validation_generator_res =
train_val_generators(train_dir, validation_dir, (224, 224))
```

Result Predicted Gender

hasilnya (waktu, steps, akurasi, val_akurasi):

```
418s 4s/step 0.5337 0.7732
497s 5s/step 0.5648 0.7512
644s 6s/step 0.6666 0.5878
719s 7s/step 0.7181 0.7512
719s 7s/step 0.7382 0.7695
716s 7s/step 0.7479 0.8317
728s 7s/step 0.6341 0.6232
710s 7s/step 0.7325 0.7634
711s 7s/step 0.7688 0.8317
735s 7s/step 0.7790 0.7293
```

Hasil Evaluasi

1. **Hasil Augmentasi mengalami penurunan nilai Validasi Akurasi hingga 0,5878 & 0,6232**

Continous Improvement

1. Meningkatkan Akurasi ResNet-50 & ResNet-101 (1/2)

Parameter	VGG-16	VGG-19	ResNet-50	ResNet-101	GoogleNet-V3	Max	Min	Average
1. Accuracy	0,9975	0,9705	0,7949	0,8306	0,9674	0,9775	0,7949	0,9082
2. Val.Accuracy	0,8960	0,8700	0,7380	0,7520	0,8570	0,8960	0,7380	0,8226

2. Merubah Environment dari CPU ke GPU

-ResNet-50

```
import os
os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'
from tensorflow.keras.applications import ResNet50V2
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense,
GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam


base_model = ResNet50V2(weights='imagenet',
include_top=False, input_shape=(128, 128, 3))

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(1, activation='sigmoid')(x)
```

Epoch	Accuracy	Val.Accuracy	Train Time (s)	Train Speed (ms/step)
1	0,7673	0,8780	15	32
2	0,9753	0,8960	4	28
3	0,9883	0,8940	3	28
4	0,9983	0,8940	3	28
5	0,9998	0.8960	4	28
6	1,0000	0,9020	4	28
7	1,0000	0,9040	4	28
8	1,0000	0.9030	3	28
9	1,0000	0,9030	4	28
10	1,0000	0,9040	4	28
Max	1,0000	0,9040	15	32
Min	0,7673	0,8780	3	28
Average	0,9729	0,8969	5	28

Hasil Evaluasi

```
1/1 [==============================] - 0s 419ms/step
File: cew1.jpeg, Predicted Gender: Female
File: cew2.jpg, Predicted Gender: Female
File: cew3.jpeg, Predicted Gender: Female
File: cew4.jpg, Predicted Gender: Female
File: cew5.jpeg, Predicted Gender: Female
File: cow1.jpg, Predicted Gender: Male
File: cow2.jpeg, Predicted Gender: Male
File: cow3.jpeg, Predicted Gender: Male
File: cow4.jpg, Predicted Gender: Male
File: cow5.jpeg, Predicted Gender: Male

Total prediction time: 0.4479053020477295 seconds
```

Inference Time : 0,04479053020477295 S

Indonesia AI
AI for Everyone, AI for Indonesia

Continous Improvement

1. Meningkatkan Akurasi ResNet-50 & ResNet-101 (2/2)

Parameter	VGG-16	VGG-19	ResNet-50	ResNet-101	GoogleNet-V3	Max	Min	Average
1. Accuracy	0,9975	0,9705	0,7949	0,8306	0,9674	0,9775	0,7949	0,9082
2. Val.Accuracy	0,8960	0,8700	0,7380	0,7520	0,8570	0,8960	0,7380	0,8226

2. Merubah Environment dari CPU ke GPU

-ResNet-101

```
# Modelling menggunakan ResNet50 (sesuaikan)
import os
os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'
from tensorflow.keras.applications import ResNet101V2
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense,
GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam


base_model = ResNet101V2(weights='imagenet',
include_top=False, input_shape=(128, 128, 3))

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(1, activation='sigmoid')(x)
```

Epoch	Accuracy	Val.Accuracy	Train Time (s)	Train Speed (ms/step)
1	0,7713	0,8880	27	56
2	0,9815	0,8960	6	48
3	0,9990	0,8990	6	49
4	0,9998	0,9040	6	47
5	1,0000	0,9070	6	48
6	1,0000	0,9090	6	47
7	1,0000	0,9090	6	47
8	1,0000	0,9090	6	48
9	1,0000	0,9110	6	48
10	1,0000	0,9130	6	49
Max	1,0000	0,9130	27	56
Min	0,7713	0,8880	6	47
Average	0,9752	0,9045	8	49

Hasil Evaluasi

```
1/1 [==============================] - 1s 912ms/step
File: cew1.jpeg, Predicted Gender: Female
File: cew2.jpg, Predicted Gender: Female
File: cew3.jpeg, Predicted Gender: Female
File: cew4.jpg, Predicted Gender: Female
File: cew5.jpeg, Predicted Gender: Female
File: cow1.jpg, Predicted Gender: Male
File: cow2.jpeg, Predicted Gender: Male
File: cow3.jpeg, Predicted Gender: Male
File: cow4.jpg, Predicted Gender: Male
File: cow5.jpeg, Predicted Gender: Female

Total prediction time: 0.9398868083953857 seconds
```

Inference Time : 0,09398868083953857 S

1. **Meningkatkan Akurasi ResNet-50 & ResNetNet-101 dicoba dilakukan 2 Metode yaitu melakukan Augmentasi Data Image dan Merubah Environment dari CPU ke GPU.**
2. **Hasil Akurasi ResNet-50 & ResNet-101 memiliki hasil akurasi lebih baik hingga mencapai 100% dan Validasi Akurasi >90% ketika Merubah Merubah Environment dari CPU ke GPU.**

Indonesia AI

AI for Everyone, AI for Indonesia