

Project 1

Gender Classification

Vision Plus





The Team

Teguh Imanto

Andre Fransiscus Masalle

Christian Bernard Kuswandi

Robert Rasidy

Salsabylla Nada Apsariny

Abraham Bob Ellson



Background & Problem Statement

Background

Pengenalan wajah (face recognition) merupakan salah satu aplikasi penting dalam kecerdasan buatan, dengan penggunaan luas di bidang keamanan, pemasaran, dan interaksi manusia-komputer.

Salah satu fungsinya adalah **klasifikasi gender** berdasarkan citra wajah, yang dapat memberikan nilai tambah dalam analisis demografi atau sistem personalisasi.

Banyak bisnis ingin tahu profil demografis pengguna mereka (usia, gender, dll) untuk **menyajikan konten, iklan, atau produk yang lebih relevan**. Contoh: Aplikasi belanja online, e-commerce, atau media sosial bisa menyesuaikan rekomendasi produk berdasarkan gender.

Gender classification digunakan untuk **statistik pengguna**, Misalnya, dalam survei publik, aplikasi, atau event, kita bisa tahu berapa banyak user pria/wanita yang menggunakan layanan. Beberapa sistem keamanan memerlukan identifikasi cepat, termasuk gender.
Smart surveillance: Mendeteksi pria/wanita di area publik.

Background & Problem Statement

Problem Statement

Meskipun klasifikasi gender tampak sederhana bagi manusia, sistem komputer sering kali menghadapi tantangan seperti pencahayaan tidak konsisten, ekspresi wajah, pose, dan kemiripan visual antar gender.

Objectives & Scope

Objective

Membangun model klasifikasi gender (laki-laki dan perempuan) dari citra wajah dengan melakukan eksperimen menggunakan tiga algoritma deep learning populer, yaitu VGG19, GoogLeNet, ResNet50 dan ResNet18.

Mencakup proses tuning hyperparameter untuk mengoptimalkan performa masing-masing model, serta melakukan evaluasi dan analisis perbandingan guna menentukan algoritma yang memberikan hasil terbaik.

Scope

1. Mengimplementasikan tiga arsitektur CNN sebagai model klasifikasi: VGG19, GoogLeNet, ResNet50, ResNet18.
2. Melatih dan menguji model untuk dua kelas saja: laki-laki dan perempuan.
3. Tidak mencakup klasifikasi usia, ekspresi wajah, atau identifikasi individu (face recognition berdasarkan identitas unik).

Sample Images

Female



Male



Data Collection

DataSet menggunakan Large-scale CelebFaces Attributes (CelebA) Dataset

<https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>

CelebFaces Attributes Dataset (CelebA) adalah kumpulan data atribut wajah berskala besar dengan lebih dari **200 ribu** gambar selebriti, masing-masing dengan **40** anotasi atribut.

- **10.177** jumlah **identitas** ,
- **202.599** jumlah **gambar wajah** , dan
- **5 lokasi penting** , **40 anotasi atribut biner** per gambar.

Lalu disederhanakan menjadi 5000 gambar

<https://drive.google.com/drive/folders/1BMcCUkpRA99ULUMWYUObol19Yph2k3Vs>

Data Attribute

```
[47]: ['5_o_Clock_Shadow',  
      'Arched_Eyebrows',  
      'Attractive',  
      'Bags_Under_Eyes',  
      'Bald',  
      'Bangs',  
      'Big_Lips',  
      'Big_Nose',  
      'Black_Hair',  
      'Blond_Hair',  
      'Blurry',  
      'Brown_Hair',  
      'Bushy_Eyebrows',  
      'Chubby',  
      'Double_Chin',  
      'Eyeglasses',  
      'Goatee',  
      'Gray_Hair',  
      'Heavy_Makeup',  
      'High_Cheekbones',  
      'Male',  
      'Mouth_Slightly_Open',  
      'Mustache',  
      'Narrow_Eyes',  
      'No_Beard',  
      'Oval_Face',  
      'Pale_Skin',  
      'Pointy_Nose',  
      'Receding_Hairline',  
      'Rosy_Cheeks',  
      'Sideburns',  
      'Smiling',  
      'Straight_Hair',  
      'Wavy_Hair',  
      'Wearing_Earrings',  
      'Wearing_Hat',  
      'Wearing_Lipstick',  
      'Wearing_Necklace',  
      'Wearing_Necktie',  
      'Young']
```

```
[45]: data_lines[0]
```

```
[45]: '000001.jpg -1 1 1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 1 1 -1 1 -1 -1 1 -1 -1 1 1 -1 1 -1 1 -1 -1 1\n'
```

Data Attribute

Cek Jumlah Data

```
[48]: # Parsing data di list_attribute.txt
data = []
for line in data_lines:
    parts = line.strip().split()
    filename = parts[0]
    attributes = [int(x) for x in parts[1:]]
    data.append([filename] + attributes)

columns = ['filename'] + header
df_attributes = pd.DataFrame(data, columns=columns)

# Filter gambar yang ditemukan di image folder dan `filename` dari df_attributes
existing_files = []
for idx, row in df_attributes.iterrows():
    filename = row['filename']
    image_path = os.path.join(image_folder, filename)
    if os.path.exists(image_path):
        existing_files.append(idx)

df_attributes = df_attributes.iloc[existing_files].reset_index(drop=True) # Filter

print(f"Total samples: {len(df_attributes)}")
print(f"Male count: {(df_attributes['Male'] == 1).sum()}")
print(f"Female count: {(df_attributes['Male'] == -1).sum()}")

Total samples: 5000
Male count: 2047
Female count: 2953
```


Data Preparation

1. Duplicated Data & Eleminate

```
File 182912(1).jpg dihapus.  
File 183121(1).jpg dihapus.  
File 182809(1).jpg dihapus.  
File 189513(1).jpg dihapus.  
File 189651(1).jpg dihapus.  
File 182943(1).jpg dihapus.  
File 183018(1).jpg dihapus.  
File 183050(1).jpg dihapus.  
File 189581(1).jpg dihapus.  
File 189132(1).jpg dihapus.  
File 182793(1).jpg dihapus.  
File 183145(1).jpg dihapus.  
File 189297(1).jpg dihapus.  
File 189324(1).jpg dihapus.  
File 183005(1).jpg dihapus.  
File 183111(1).jpg dihapus.  
File 189512(1).jpg dihapus.
```

Selesai menghapus file duplikat.

Jumlah gambar dalam folder: 5000

2. Ekstrak Label & Filter

```
[159]:
```

	images	Male
0	000051.jpg	1
1	000052.jpg	1
2	000065.jpg	1
3	000166.jpg	1
4	000198.jpg	0
...
4995	202320.jpg	0
4996	202340.jpg	0
4997	202347.jpg	0
4998	202357.jpg	0
4999	202566.jpg	1

5000 rows × 2 columns

3. Split Data

```
# Split data 80:20  
train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)
```

Data Preprocessing

Augmentation Data

- **`transforms.Resize((224, 224))`** > Mengubah ukuran gambar 224x224 pixel
- **`transforms.ToTensor()`** > Mengubah format gambar (PIL Image atau NumPy array) menjadi PyTorch Tensor & skala pixelnya diubah dari 0-255 menjadi 0-1
- **`transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),])`** > Menormalisasi nilai pixel dengan mean dan standard deviation (Ini dilakukan supaya data input cocok dengan model pretrained (misalnya ResNet di ImageNet))

Model Development

Model Architecture

1. `resnet50 = resnet50(weights=ResNet50_Weights.DEFAULT)`
2. `googlenet = models.googlenet(pretrained=True)`
3. `vgg19 = vgg19(weights=VGG19_Weights.IMAGENET1K_V1)` (**Freeze Layer**)
4. `resnet18 = resnet18(weights=ResNet18_Weights.DEFAULT)`

Note:

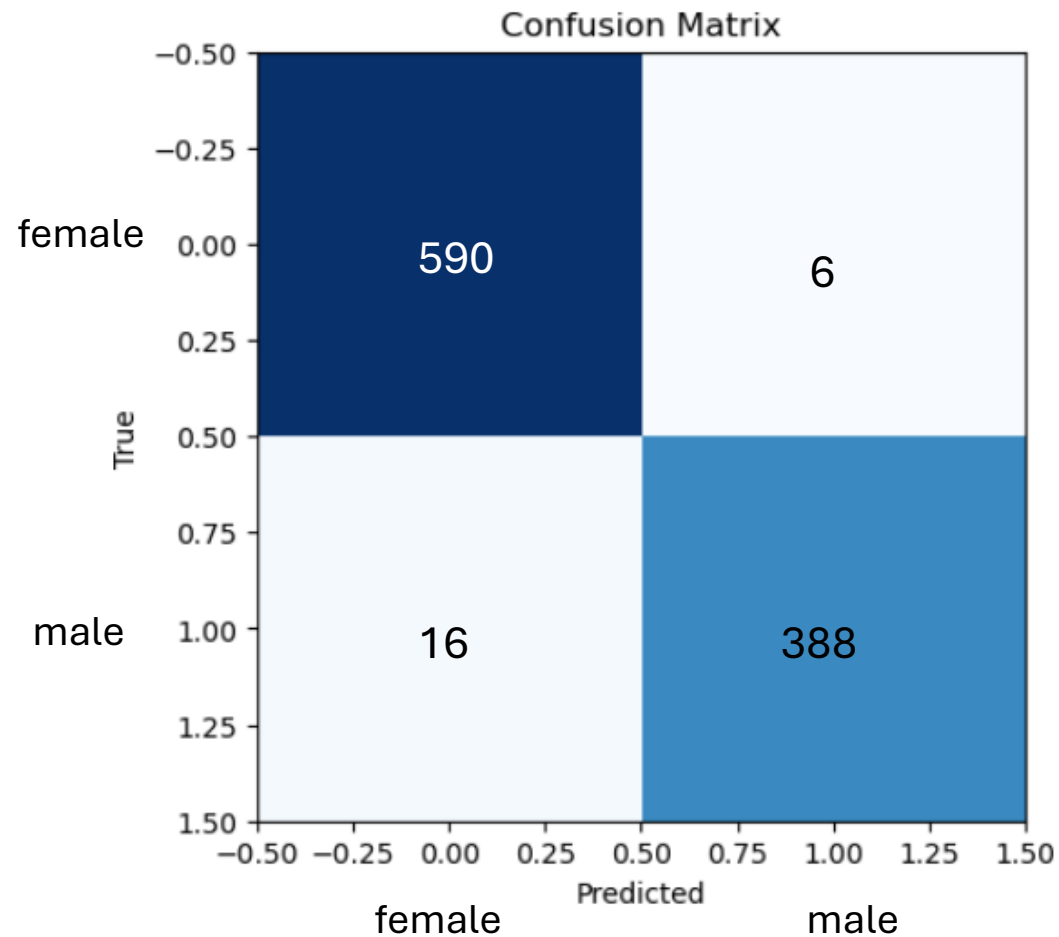
- Not Using Freeze Layer (Resnet50, GoogLeNet, ResNet18)
- Pretrained ImageNet

Hyperparameter Tuning

- Optimizer = Adam
- Learning rate = 0.001 (0.0002 for VGG19)
- Batch size = 32
- Jumlah epoch = 10 (20 for VGG19)
- Creation = CrossEntropyLoss

Model Evaluation

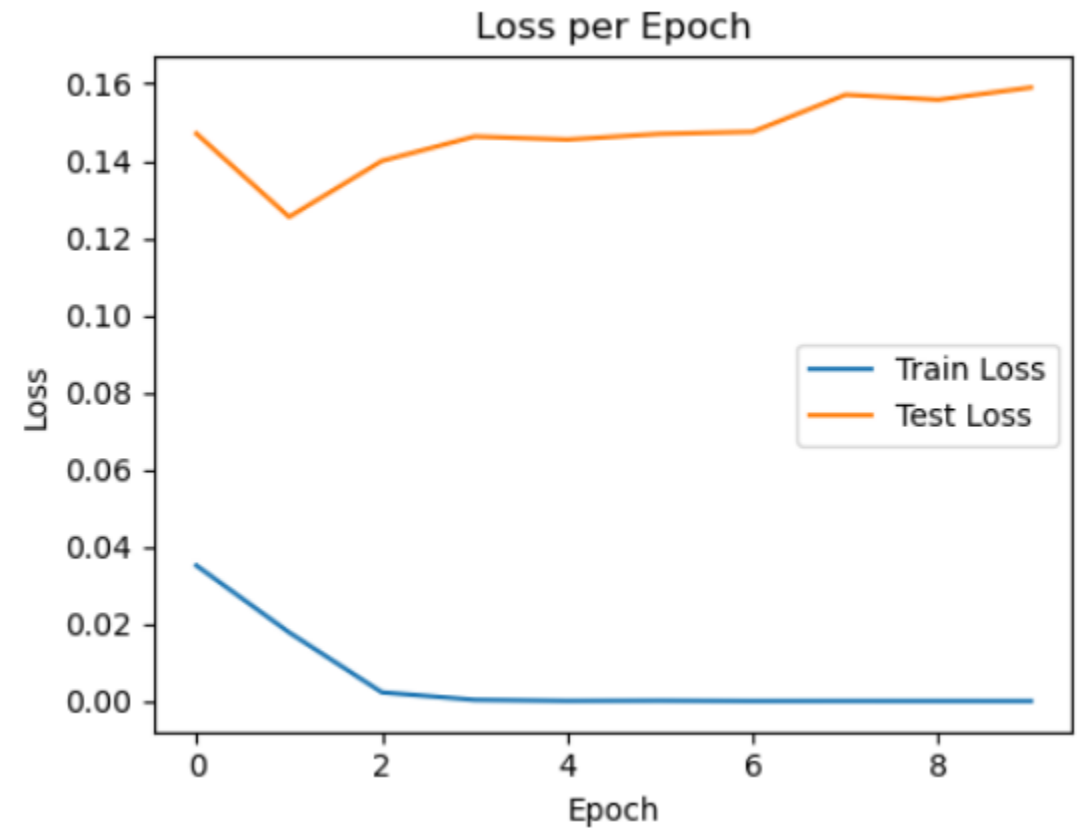
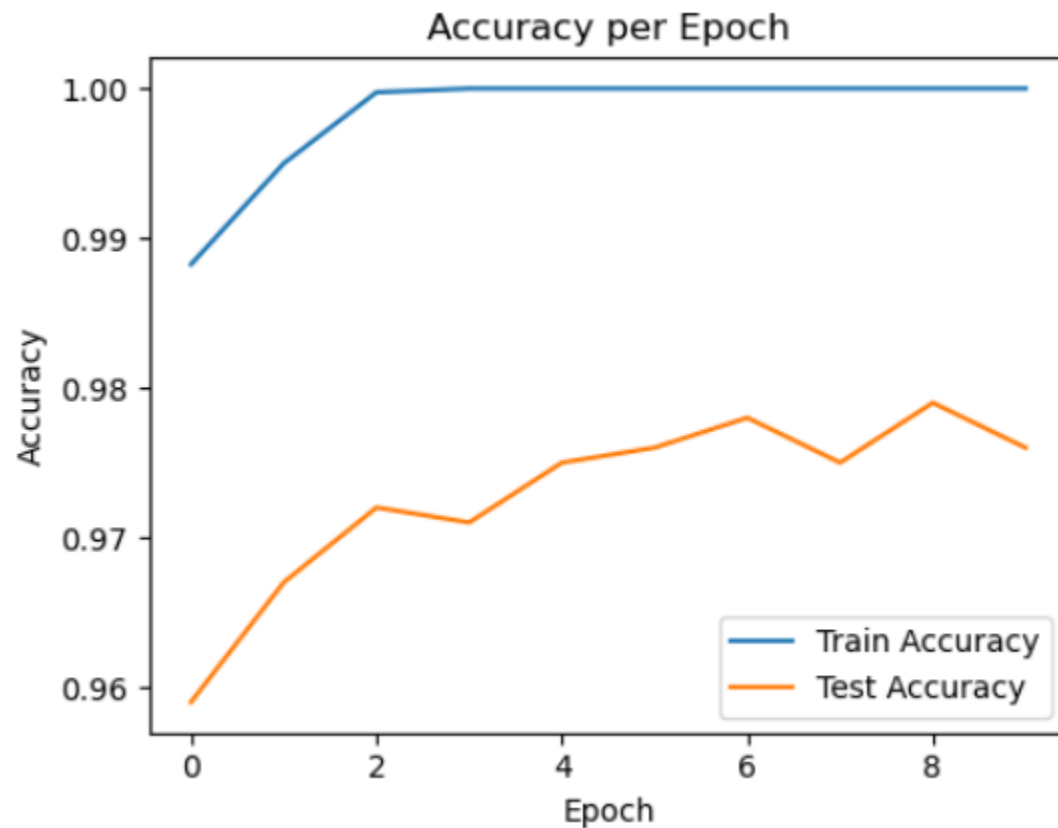
ResNet50



	precision	recall	f1-score	support
female	0.97	0.99	0.98	596
male	0.98	0.96	0.97	404
accuracy			0.98	1000
macro avg	0.98	0.97	0.97	1000
weighted avg	0.98	0.98	0.98	1000

Model Evaluation

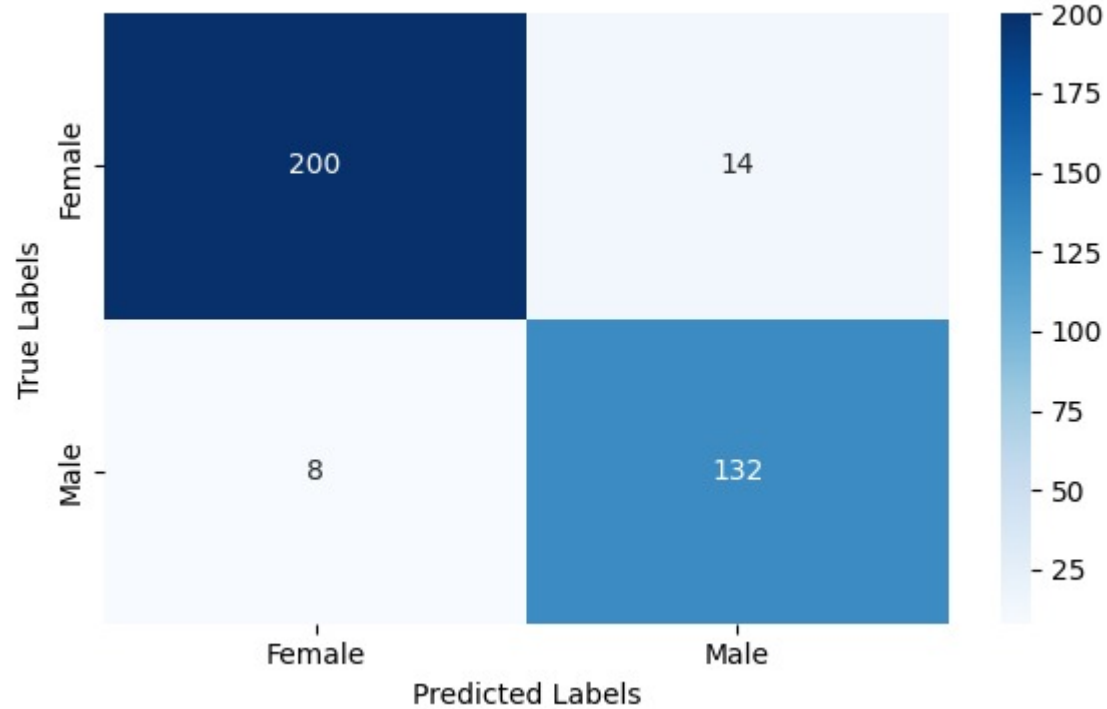
ResNet50



Model Evaluation

ResNet18

Confusion Matrix



Test Accuracy: 93.79%

Classification Report:

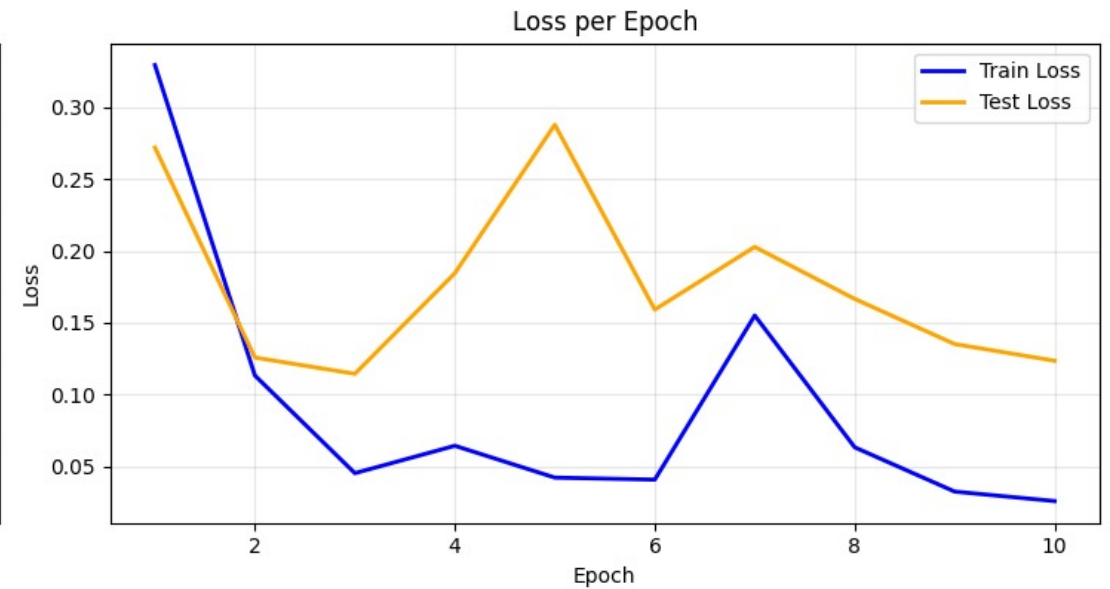
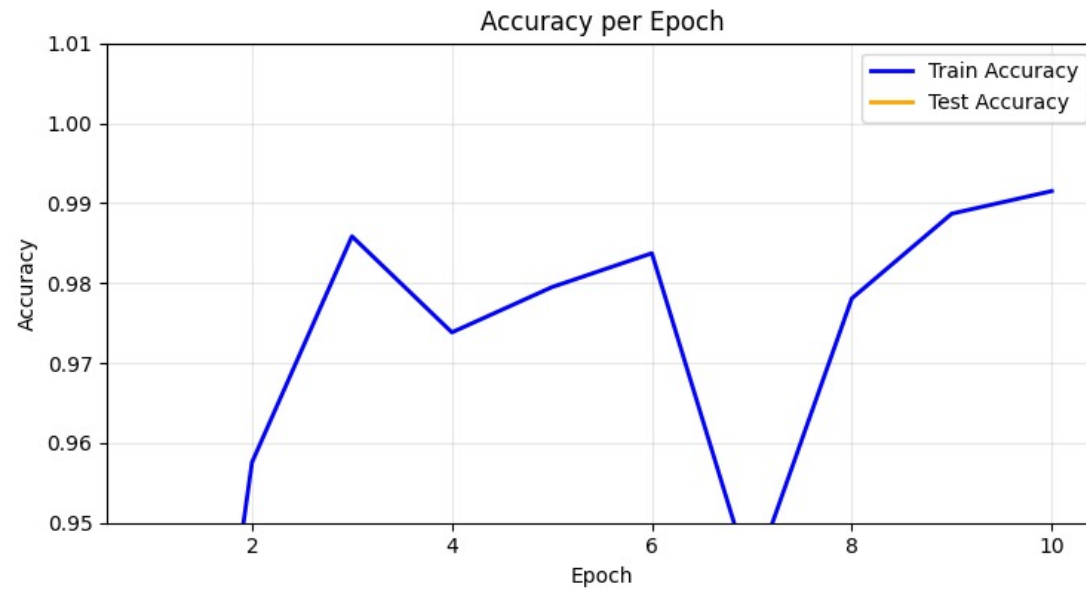
	precision	recall	f1-score	support
Female	0.96	0.93	0.95	214
Male	0.90	0.94	0.92	140
accuracy			0.94	354
macro avg	0.93	0.94	0.94	354
weighted avg	0.94	0.94	0.94	354

Model Evaluation

ResNet18

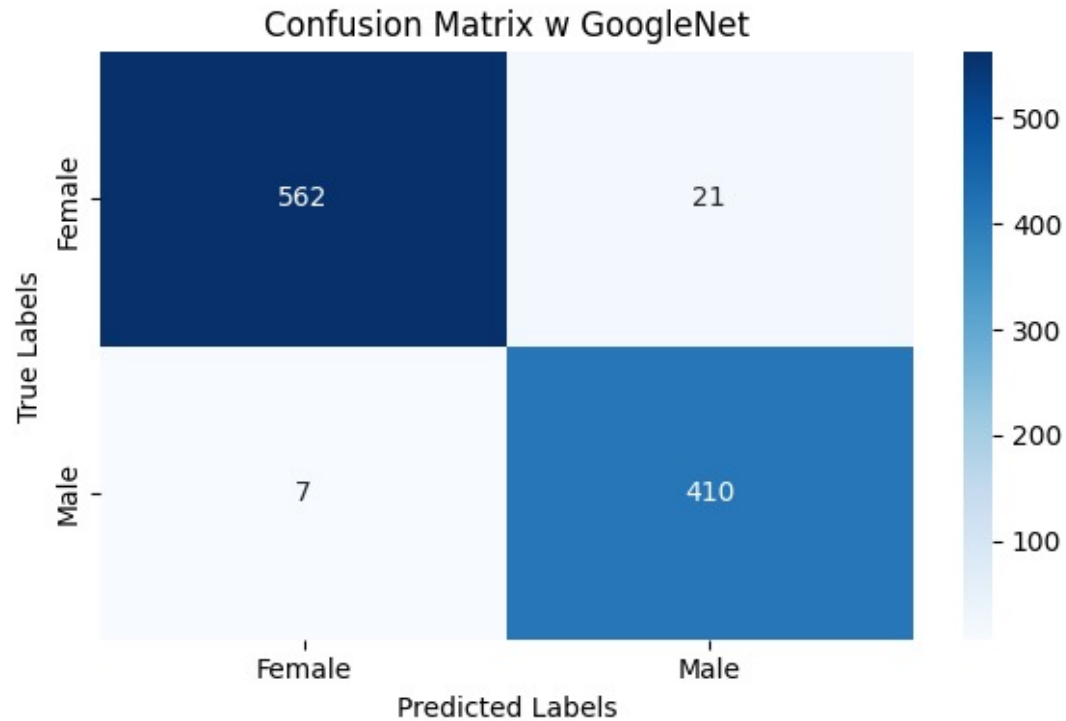
Model Evaluation

ResNet18



Model Evaluation

GoogleNet



Test Accuracy: 97.20%

Classification Report:

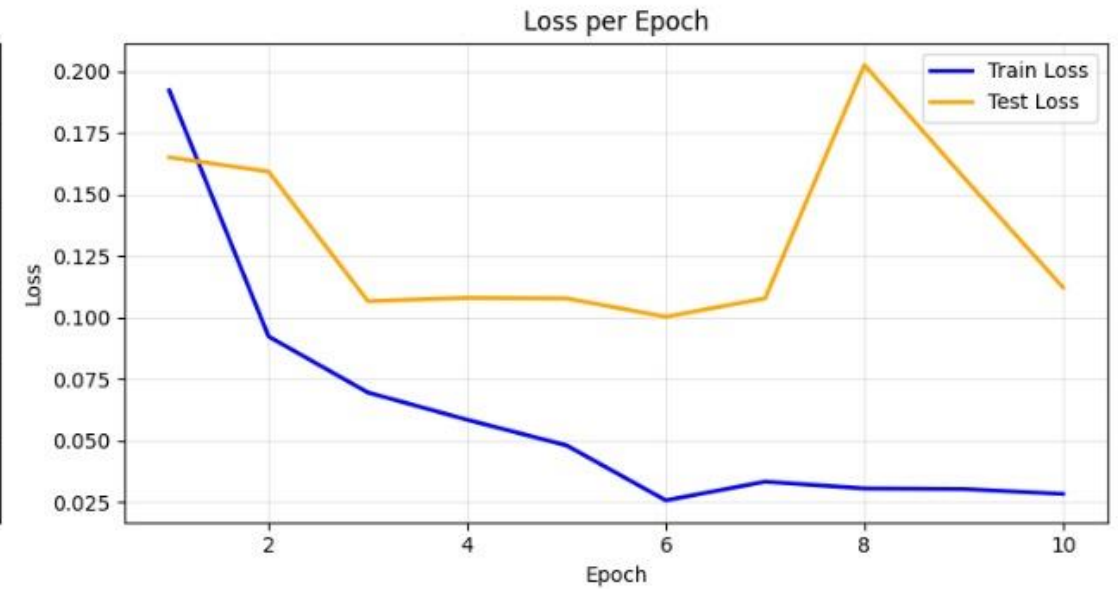
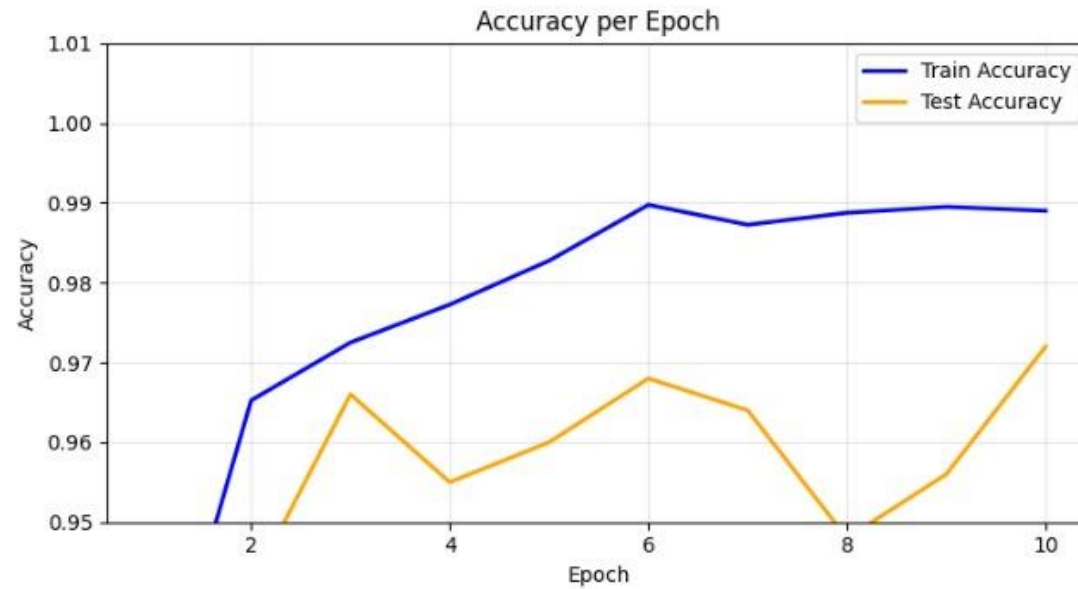
	precision	recall	f1-score	support
Female	0.99	0.96	0.98	583
Male	0.95	0.98	0.97	417
accuracy			0.97	1000
macro avg	0.97	0.97	0.97	1000
weighted avg	0.97	0.97	0.97	1000

Model Evaluation

GoogleNet

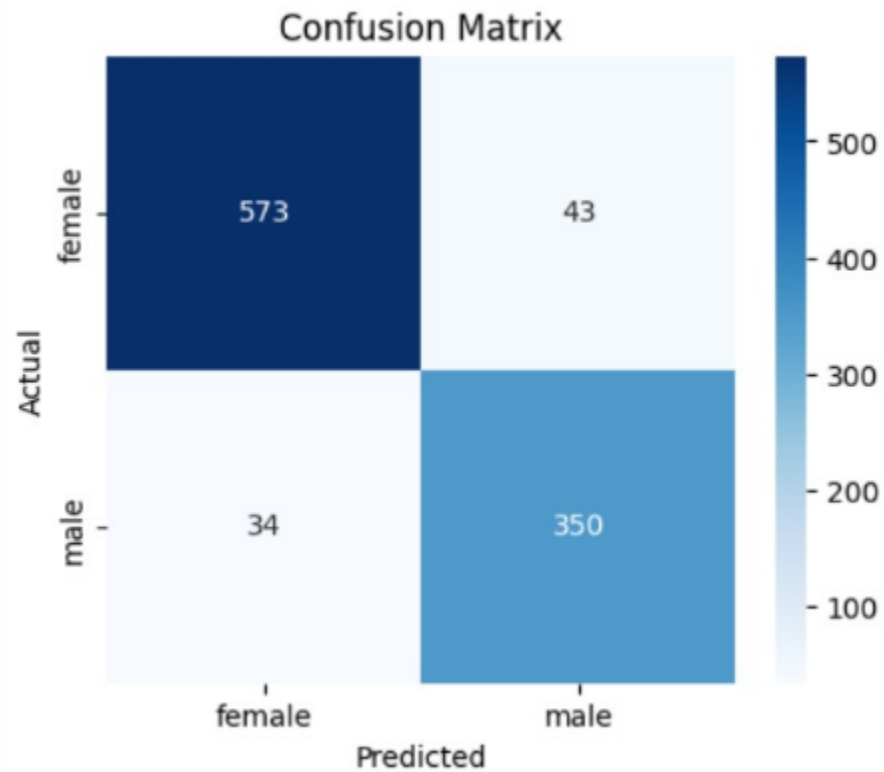
Model Evaluation

GoogleNet



Model Evaluation

VGG19



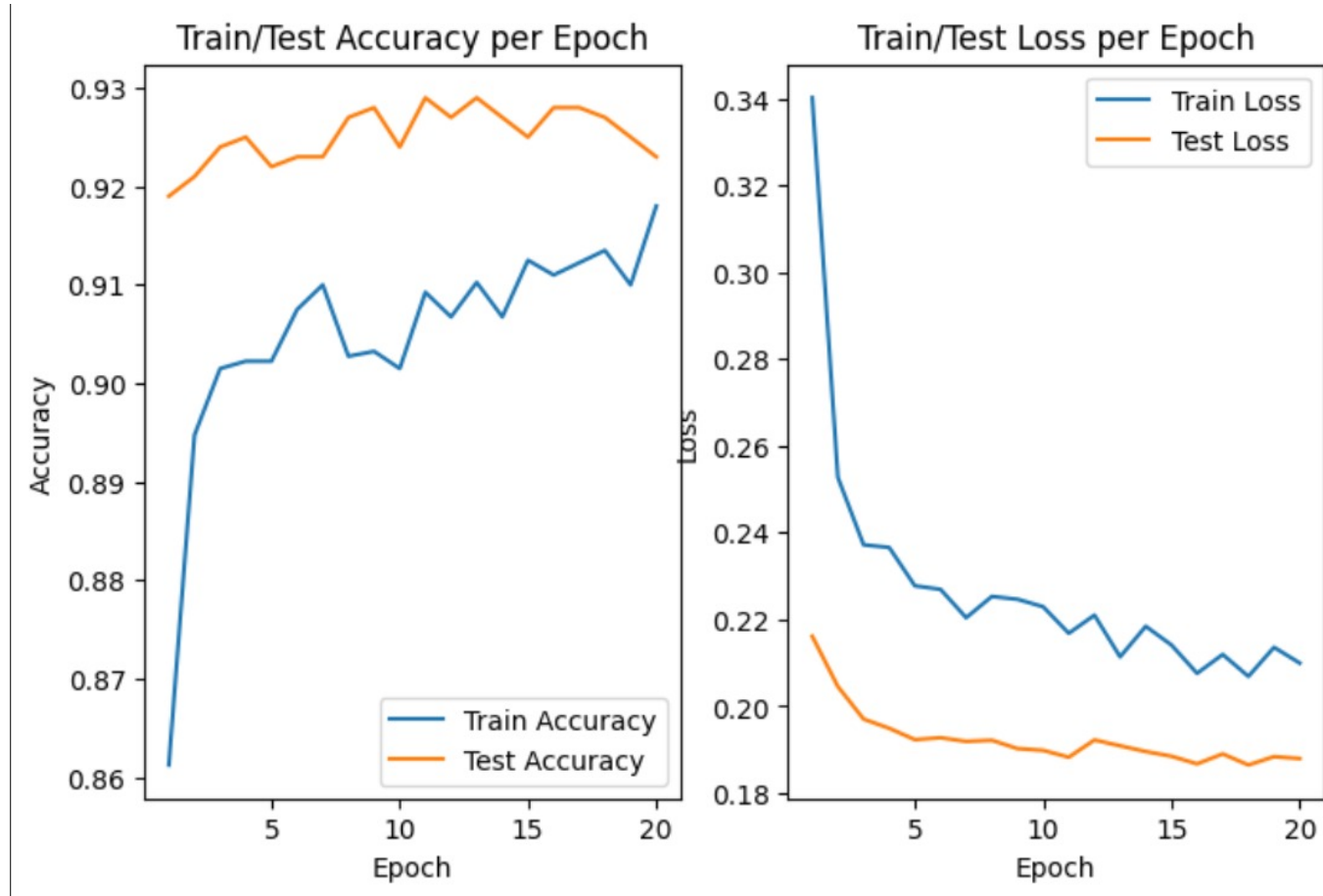
```
Classification Report:
              precision    recall  f1-score   support

   female      0.94      0.93      0.94       616
    male      0.89      0.91      0.90       384

   accuracy              0.92              1000
  macro avg              0.92              1000
 weighted avg              0.92              1000
```

Model Evaluation

VGG19



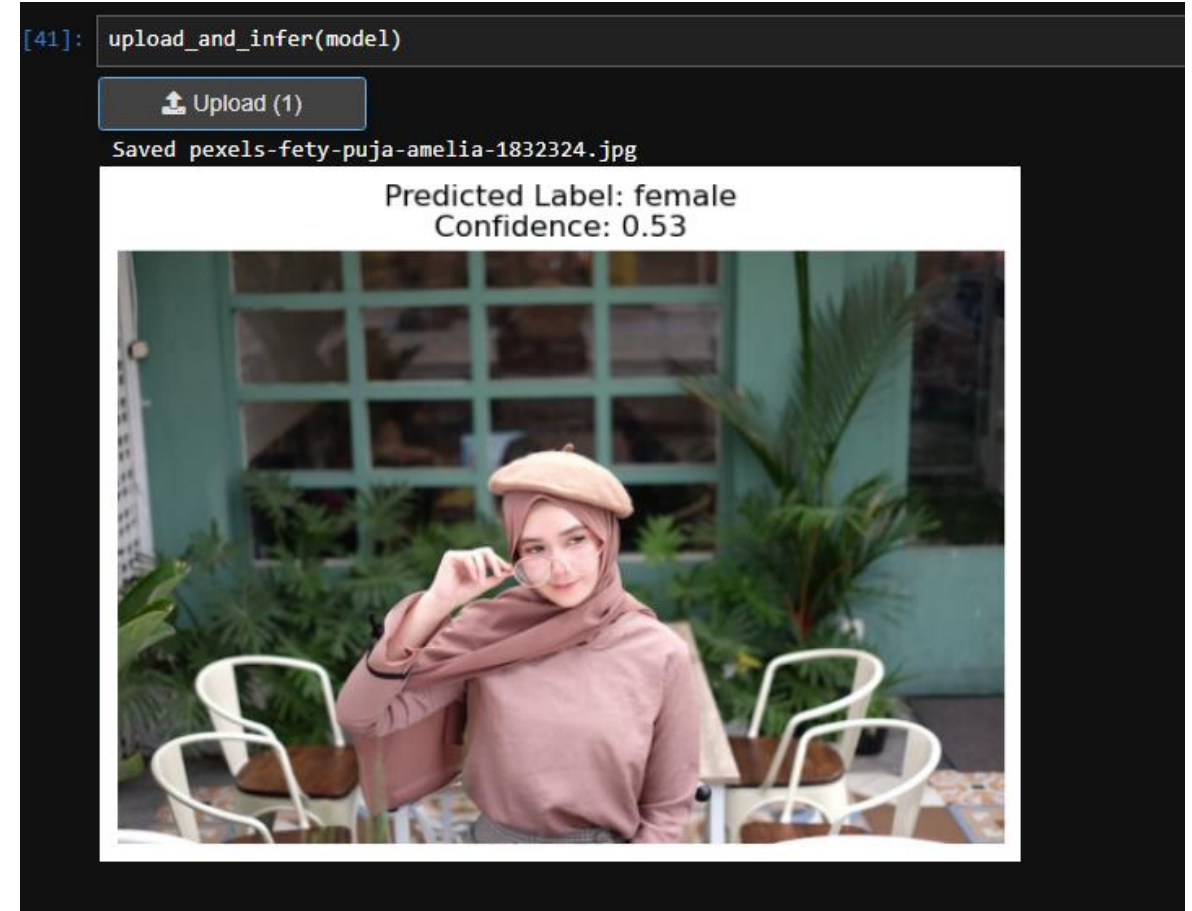
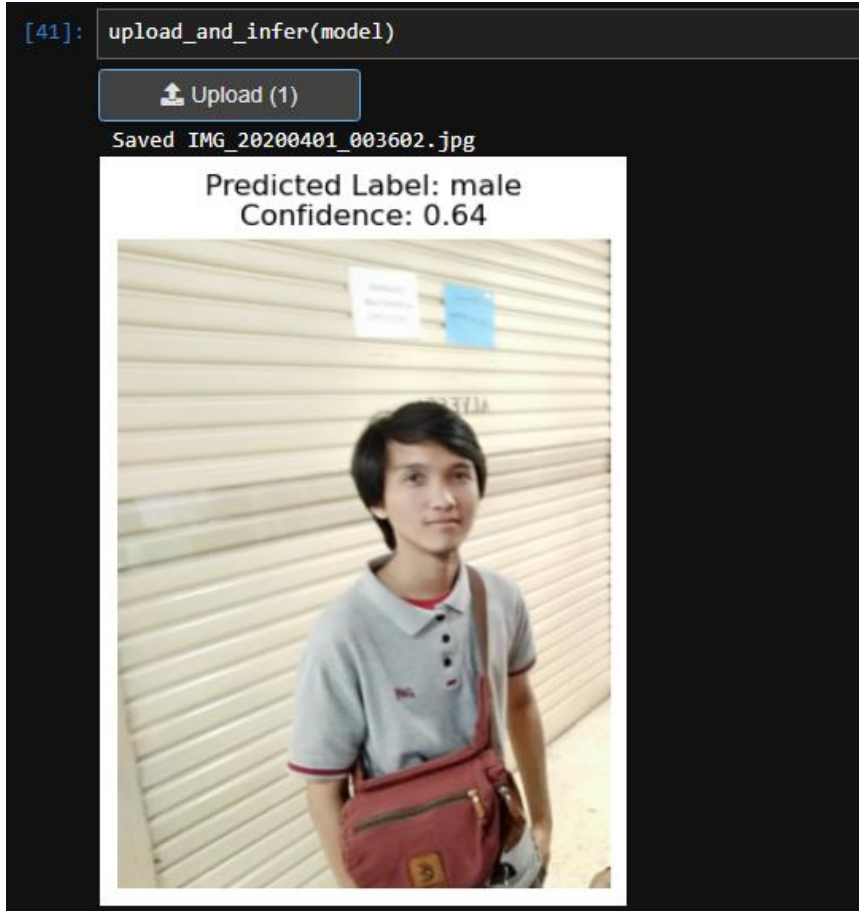
Model Comparison & Selection

Model	Accuracy	Precision (Female)	Recall (female)	f1-score (Female)	Precision (Male)	Recall (Male)	f1-score (Male)
ResNet50	0.98	0.97	0.99	0.99	0.98	0.96	0.97
GoogLeNet	0.95	0.94	0.99	0.96	0.98	0.90	0.94
VGG19	0.92	0.94	0.93	0.94	0.89	0.91	0.90
ResNet18	0.94	0.96	0.93	0.95	0.90	0.94	0.92

- ResNet50 menunjukkan performa terbaik, dengan akurasi tertinggi (98%) dan f1-score rata-rata mendekati 0.97–0.99.
- GoogLeNet memiliki akurasi yang baik (95%), tetapi recall untuk male relatif lebih rendah (0.90).
- VGG19 memberikan hasil yang cukup baik (92% akurasi), tetapi lebih rendah dibanding dua model lainnya, dengan f1-score rata-rata 0.90–0.94.

Inference Images

ResNet50



Summary

- ResNet50 menjadi pilihan model terbaik dalam eksperimen ini, menunjukkan kemampuan generalisasi yang baik untuk klasifikasi gender.
- Meskipun ResNet18, GoogLeNet dan VGG19 dapat digunakan, mereka menunjukkan keterbatasan dalam recall untuk male dan overall accuracy terhadap data test.

Future Development

- Menggunakan Full Dataset Images pada Dataset CelebA
- DataSet displit jadi 3 (train, valid, test)
- Eksperimen dengan arsitektur baru
 - EfficientNet
 - Vision Transformer (ViT)
 - ConvNeXt
- Augmentasi tambahan u/ variasi citra (pose, pencahayaan, ekspresi)
 - RandomRotation
 - RandomHorizontalFlip
 - ColorJitter
 - RandomErasing



Terima kasih