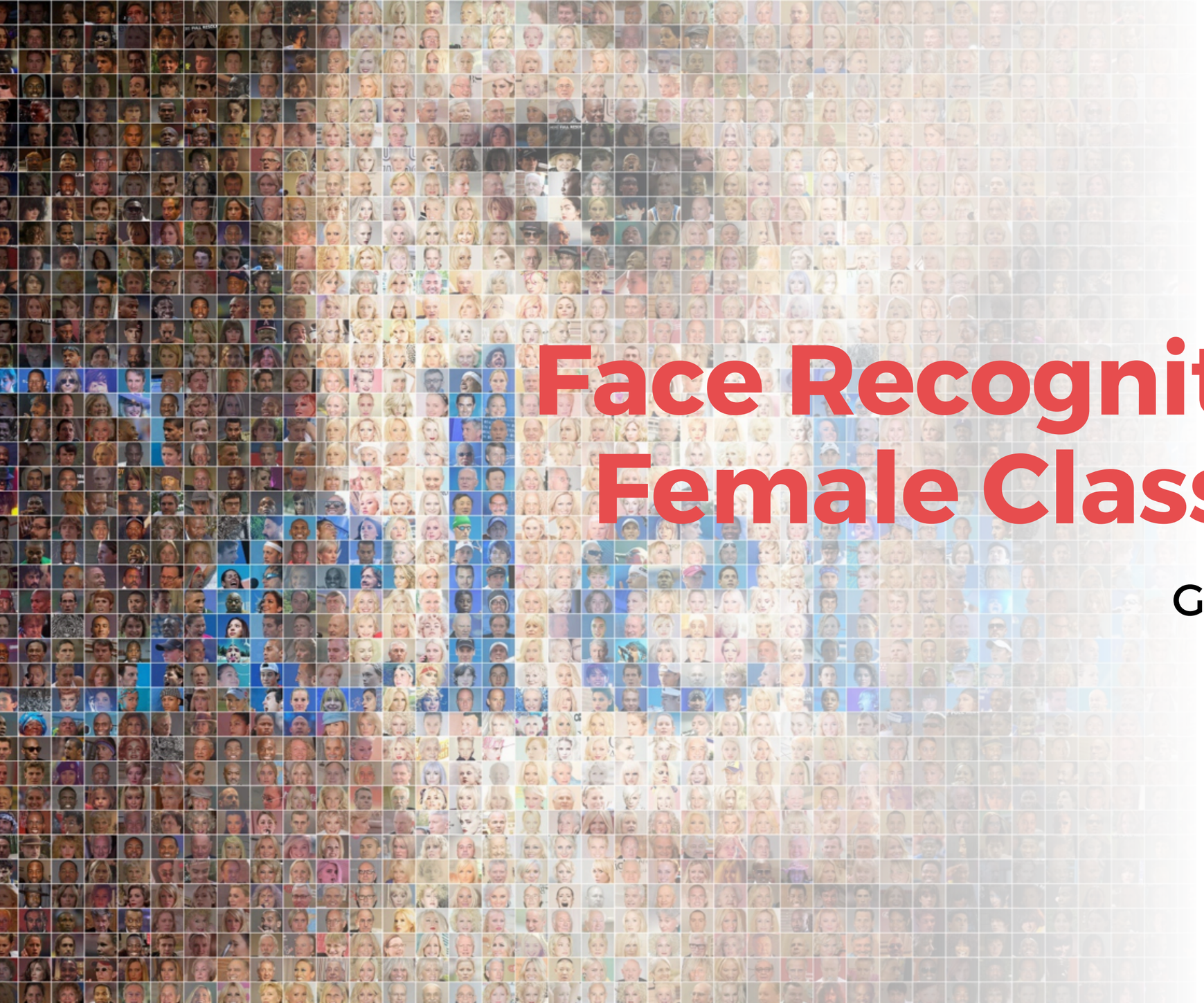


Face Recognition (Male Female Classification)

Group C - Isaac Newton

27 September 2024

Meet the Team

Dziand Dafi G.

VGG19 Experiment

Dziand Dafi Ginandjar

Zoe Muhammad

VGG16 Experiment

Aufa Biahdillah

ResNet Experiment

Aufa Biahdillah

Ryan Aprianto

GoogleNet Experiment

Ryan Aprianto

M. Fildah Rake M.

ResNet Experiment

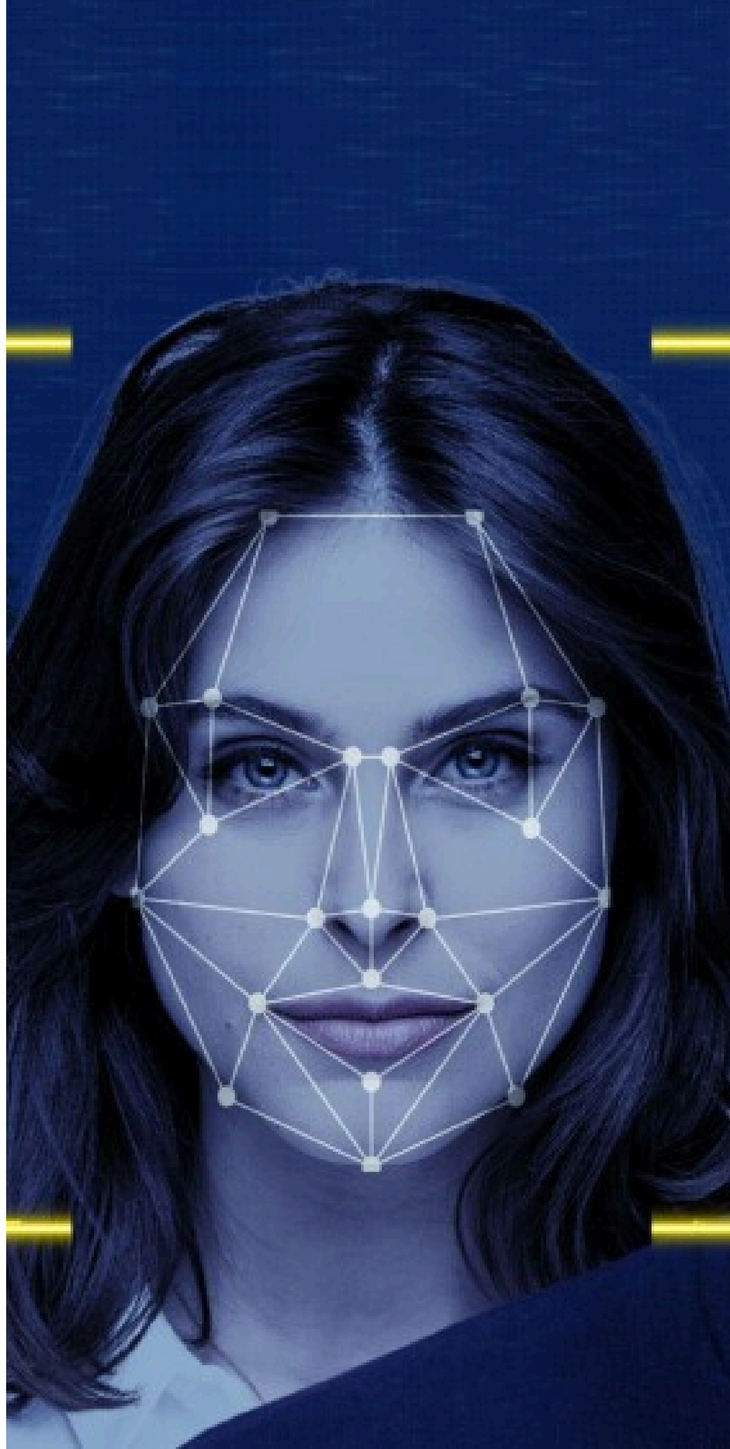
M Rizal Fauzi

Data Preparation

mochamad rizal fauzi

Background & Problem Statement

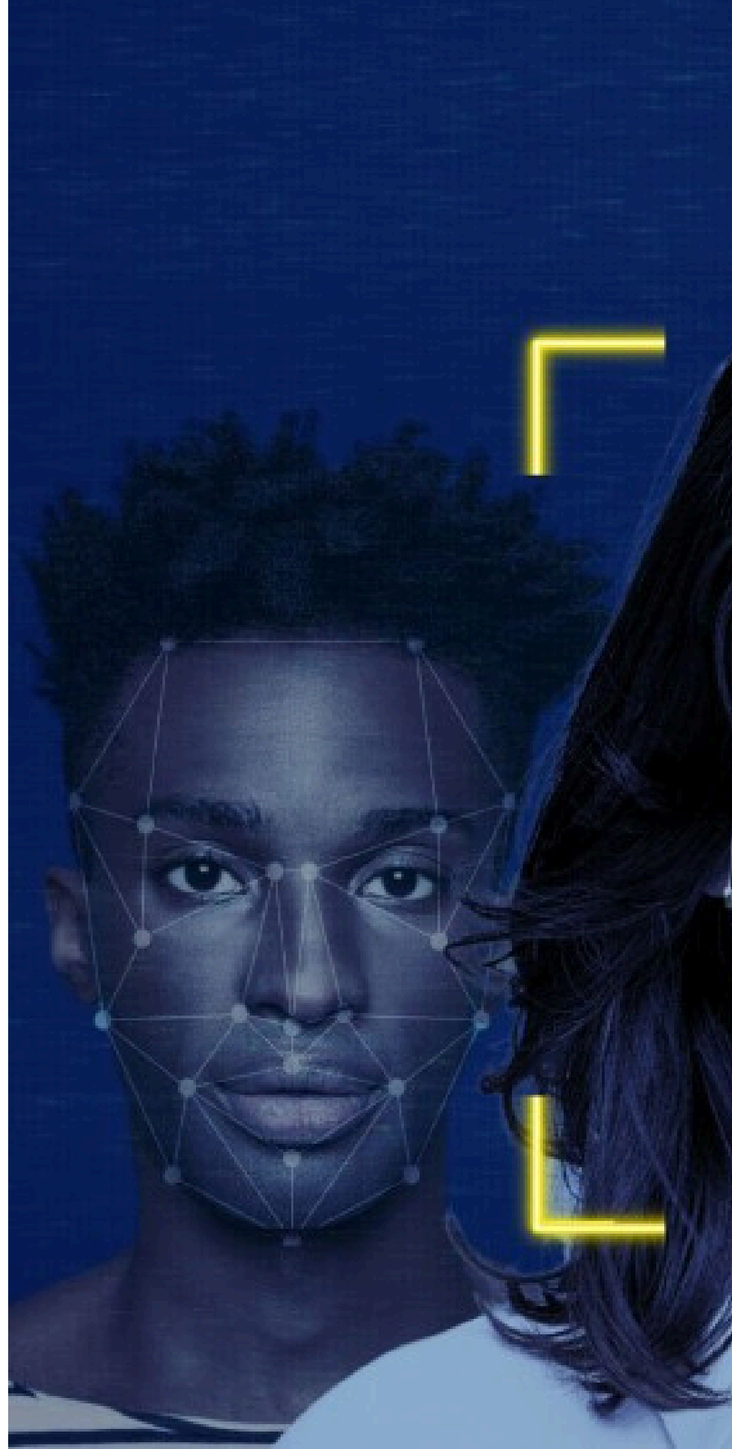
- In this digitalization era, many things can be done using gadgets and the internet.
- However, many of the tools and services are for people in general.
- Hence, to satisfy every individual needs, these tools and services need to be personalized.
- One of the solutions to overcome this issue is to implement face recognition.
- Furthermore, the advancement of technology has led to the emergence of a new problem called cybercrime.
- Again, face recognition is one way to surmount it.
- Gender classification, as one of the face recognition problem, may support this solution.



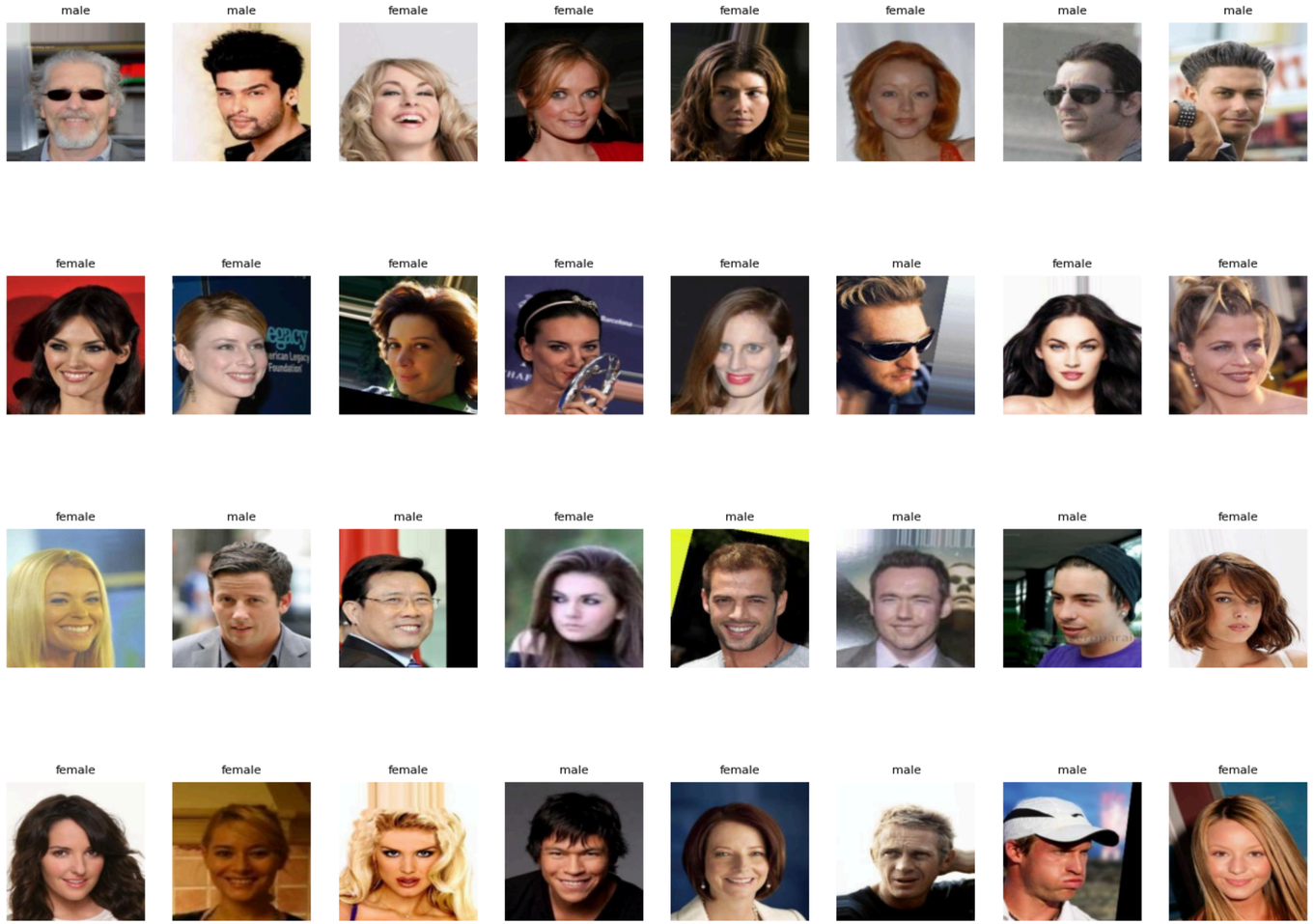
Objectives & Scope

- The objectives of this project are:
  - Conducting gender classification experiments applying three algorithms: VGG, GoogleNet, and ResNet.
  - From those experiments, the best algorithm is concluded.

- This project is limited to:
  - The dataset is from Indonesia AI, which consists of 5000 images from Celeba, including their attributes.
  - The gender is defined as male and female only.



Dataset



- The dataset from Indonesia AI is originally from Celeba dataset. They selected only 5000 images out of 202,5999 images.
- The images are the photos of some celebrities whose identities are available upon request.
- All of the images have their own attributes, one of which is the gender identity (male and female only).

Data Collection and Understanding

- Two kinds of data are being used in this project:

a. Images.zip
A zipped file of 5000 unique images from the Indonesia AI dataset.

b. list_attribute.txt
A notepad file comprising 40 kinds of labels where the gender identity is located in the 21st column. The value 1 means Male, and -1 means Female.

- There are some duplicate images in the dataset that need to be deleted before creating the model.

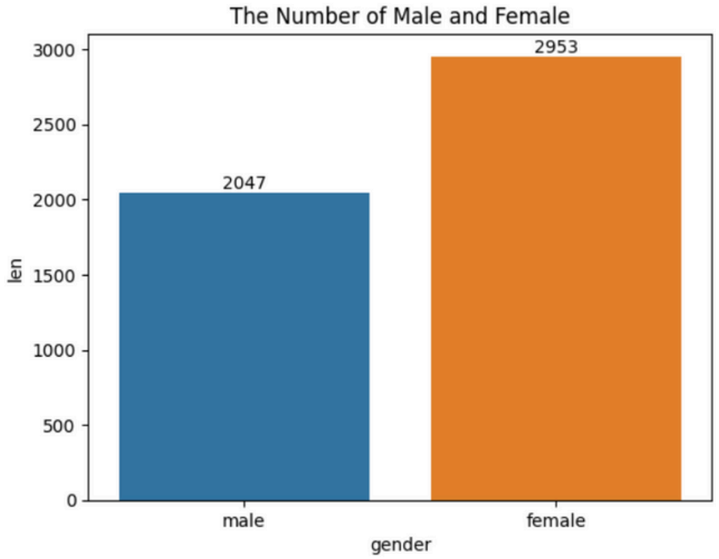
Data Cleaning

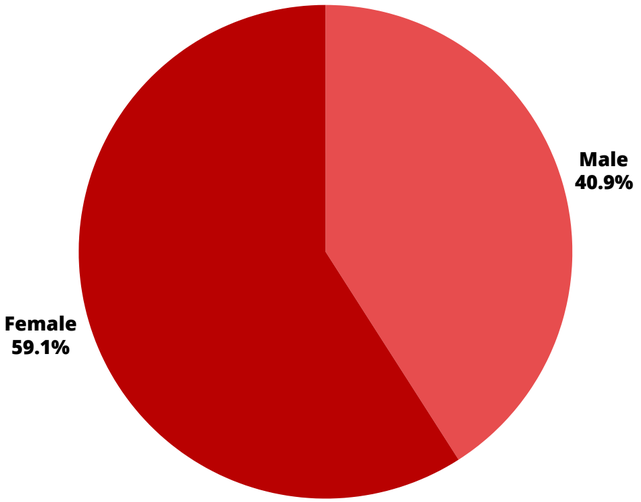
We recognized that the duplicate images have their pattern, i.e., there are extra words in their file names; for example, there is (1) in one of the file names. Thus, we clean the data as follows:

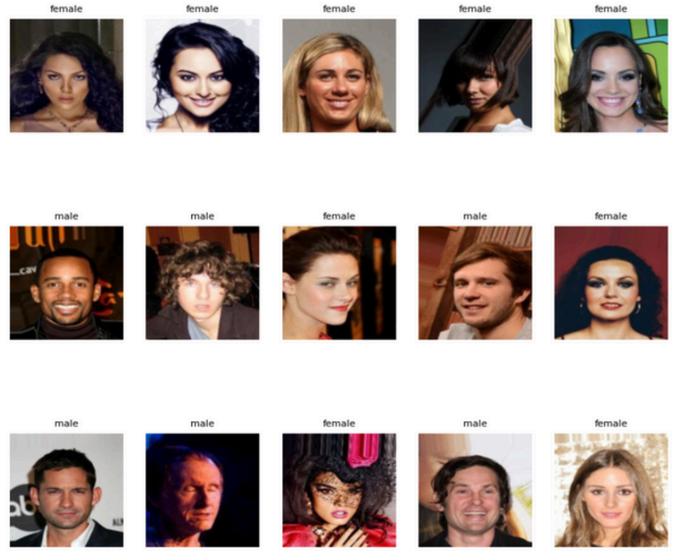
1. Extract the original file name from the filet list_attribute.txt
2. For every original file name, if the exact file name is in the folder, then we copy it to the new folder.
3. The new folder consists of male and female folders, so we filter them again based on gender attributes.
4. Finally, we removed the file from the original folder.

```
attributes = open('list_attribute.txt').readlines()

os.makedirs('/content/male', exist_ok=True)
os.makedirs('/content/female', exist_ok=True)

for line in attributes[2:]:
    clean_attr = [int(x) for x in line.split() if x in ['-1', '1']]

    gender = clean_attr[20]

    parts = line.strip().split()
    filename = parts[0].strip()

    source_path = f'/content/Images/{filename}'

    if os.path.exists(source_path):
        if gender == 1:
            target_path = f'/content/male/{filename}'
        else:
            target_path = f'/content/female/{filename}'

        print(f"File found: {source_path}")
        shutil.copy(source_path, target_path)

        os.remove(source_path)
    else:
        print(f"File not found: {source_path}")
```

Exploratory Data Analysis





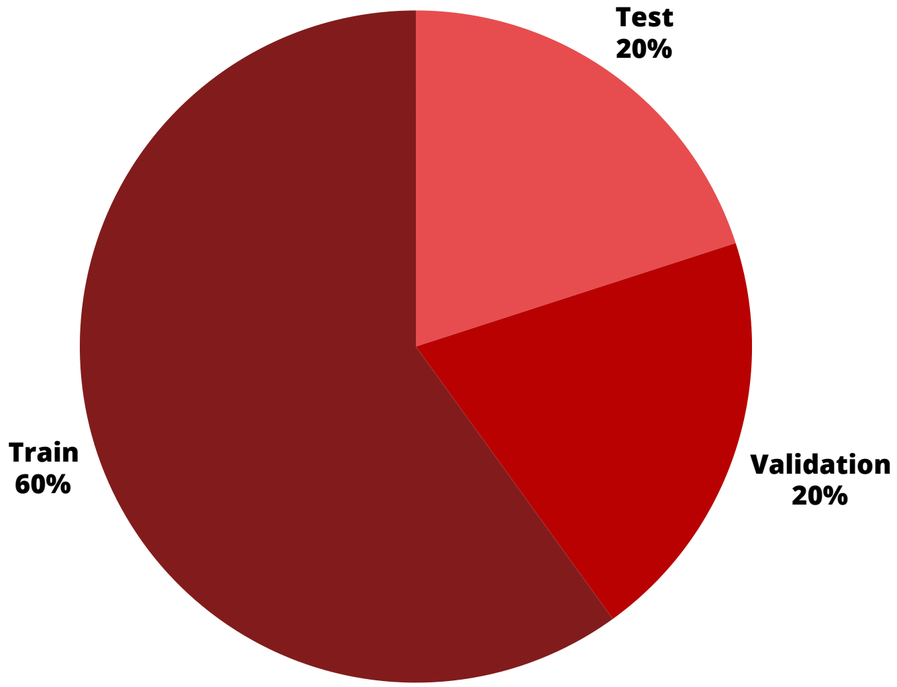


- The number of males and females in this dataset are 2047 and 2953, respectively.
- The format of the files is .jpg.
- The images are focused on the face of the individual.
- The images are in the size of 178 x 128.

Data Preparation (Train Val Test Split)

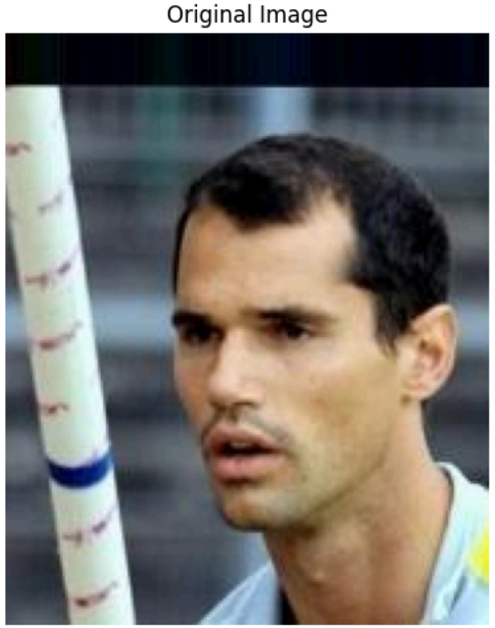
```
os.makedirs('/content/train/male', exist_ok=True)
os.makedirs('/content/val/male', exist_ok=True)
os.makedirs('/content/test/male', exist_ok=True)
os.makedirs('/content/train/female', exist_ok=True)
os.makedirs('/content/val/female', exist_ok=True)
os.makedirs('/content/test/female', exist_ok=True)

male_images = os.listdir('/content/male')
female_images = os.listdir('/content/female')

male_train_val, male_test = train_test_split(male_images, test_size=0.2, random_state=42)
male_train, male_val = train_test_split(male_train_val, test_size=0.25, random_state=42)  # 0.25

female_train_val, female_test = train_test_split(female_images, test_size=0.2, random_state=42)
female_train, female_val = train_test_split(female_train_val, test_size=0.25, random_state=42)

def move_images(file_list, source_dir, target_dir):
    for filename in file_list:
        source_path = os.path.join(source_dir, filename)
        target_path = os.path.join(target_dir, filename)
        shutil.move(source_path, target_path)

# Pindahkan gambar male ke folder yang sesuai
move_images(male_train, '/content/male', '/content/train/male')
move_images(male_val, '/content/male', '/content/val/male')
move_images(male_test, '/content/male', '/content/test/male')

# Pindahkan gambar female ke folder yang sesuai
move_images(female_train, '/content/female', '/content/train/female')
move_images(female_val, '/content/female', '/content/val/female')
move_images(female_test, '/content/female', '/content/test/female')
```

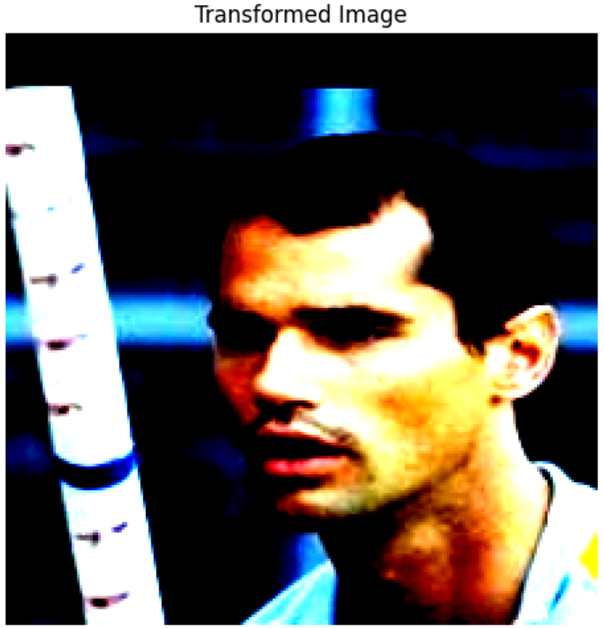
- We split the data into training, validation, and test data by using the train_test_split function with the proportion training: 2998, val: 1001, and test: 1001.



Data Preparation (Transformation)

- We apply the transformation function for every image.
- We resized the image to 224 x 224 as the models we will implement are pre-trained in this size.
- We also apply normalization for the images similar to the pre-trained model in ImageNet.






```
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Data Preparation (Data Loader)

- Then, we create the dataloaders dictionary for the train, validation, and test dataset.
- We apply batch_size = 32 to divide the data into batches with size 32 which then be trained in the model for every epoch.

```
dataloaders = {
    'train': DataLoader(image_datasets['train'], batch_size=32, shuffle=True),
    'val': DataLoader(image_datasets['val'], batch_size=32, shuffle=True),
    'test': DataLoader(image_datasets['test'], batch_size=32, shuffle=True)
}
```

Model Development and Optimization (VGG)

We finalize our model for VGG using the following settings.

VGG16
Applying the pre-trained model.

Epoch
Have 50 epochs and will be saved once the best acc found.

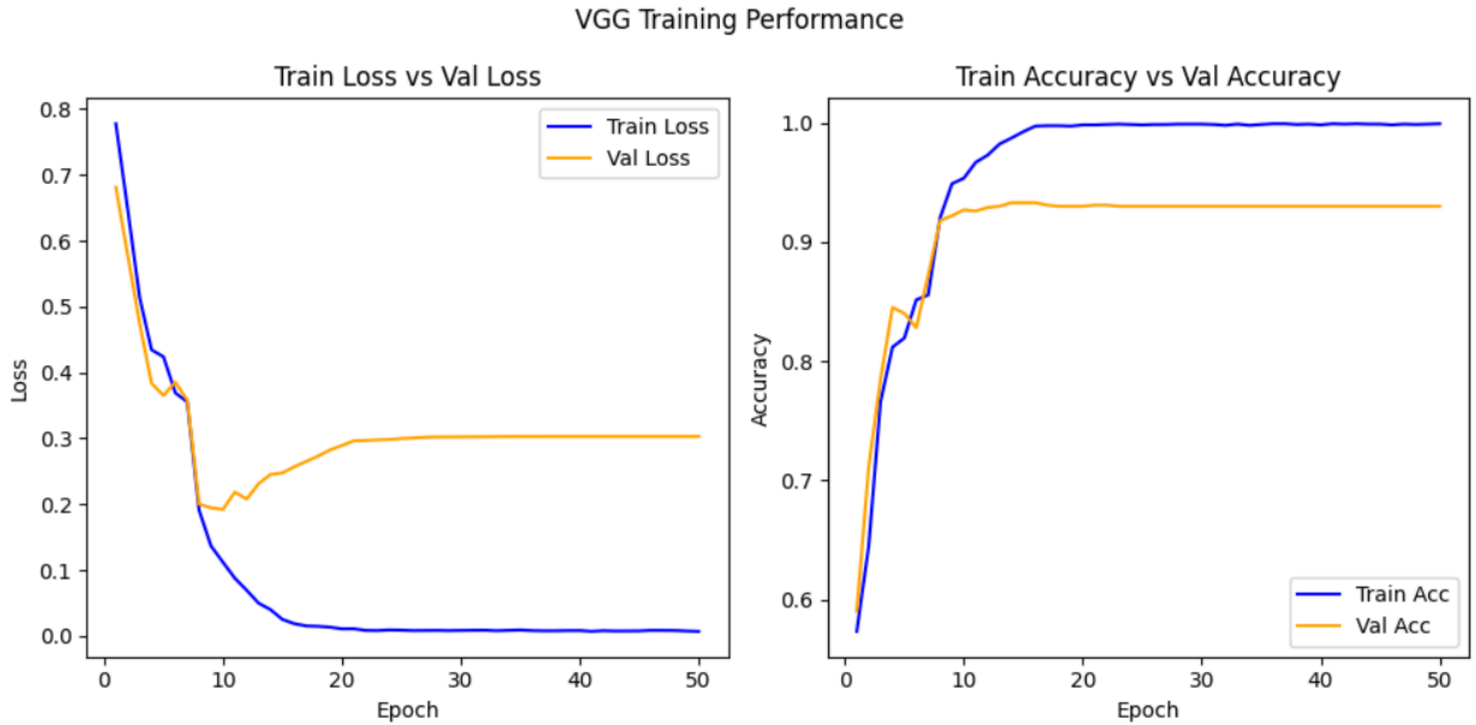
Optimizer
Adam optimizer with Lr = 0.001

FC Layer
Change the last fully connected layer into two.

Loss Function
Implementing the crossentropy loss.

Scheduler
Decrease the learning rate to optimize the model

VGG16 Performance



The best validation accuracy of the model is 0.9171 with 0.4487 loss, while the training accuracy and loss are 0.9980 and 0.0089. Since the difference is not significant, we conclude that this model is neither overfit nor underfit.

Model Development and Optimization (ResNet)

We finalize our model for ResNet using the following settings.

ResNet18
Applying the pre-trained model.

Epoch
Have 50 epochs and will be saved once the best acc found.

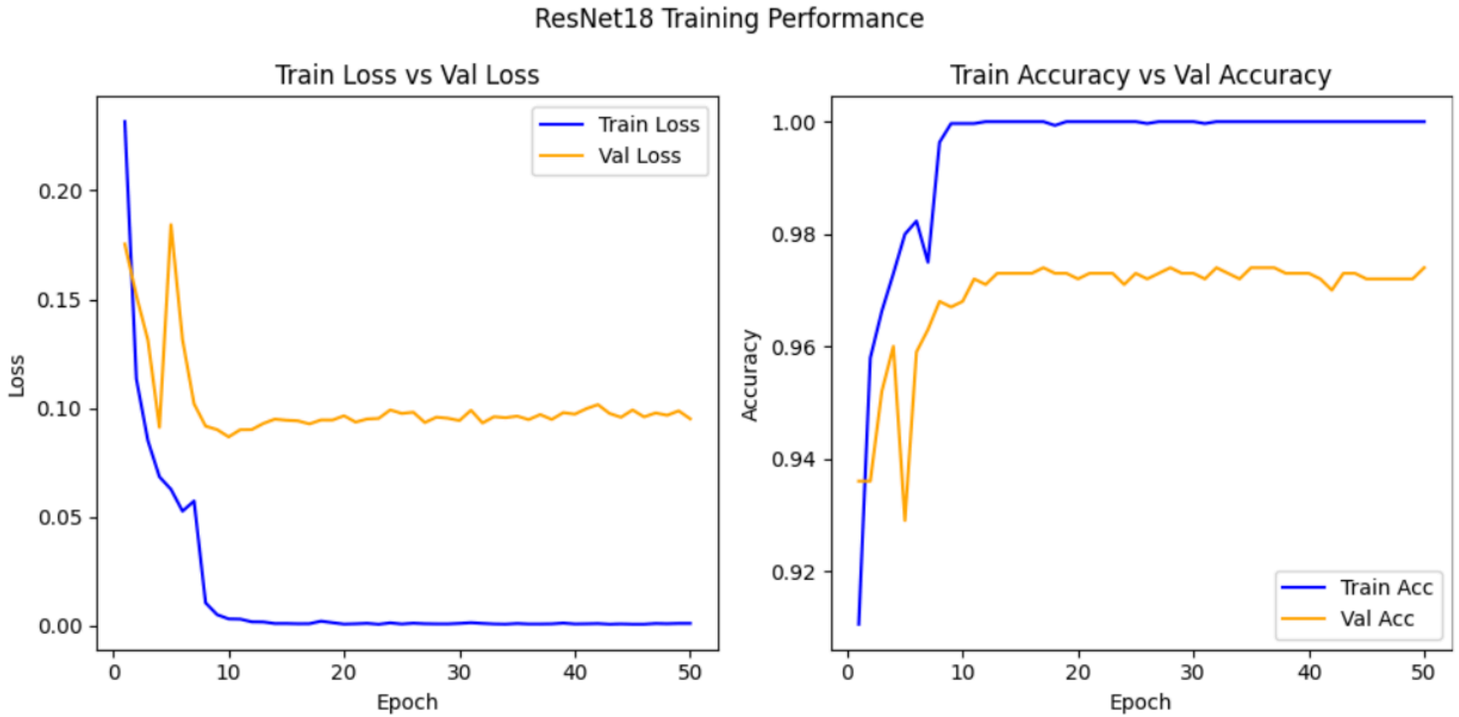
Optimizer
Adam optimizer with Lr = 0.001

FC Layer
Change the last fully connected layer into two.

Loss Function
Implementing the crossentropy loss.

Scheduler
Decrease the learning rate to optimize the model

ResNet18 Performance



The best validation accuracy of the model is 0.9760 with 0.0909 loss, while the training accuracy and loss are 1.000 and 0.0010. Since the difference is not significant, we conclude that this model is neither overfit nor underfit.

Model Development and Optimization (GoogleNet)

We finalize our model for GoogleNet using the following settings.

Inception-v1
Applying the pre-trained model.

Epoch
Have 50 epochs and will be saved once the best acc found.

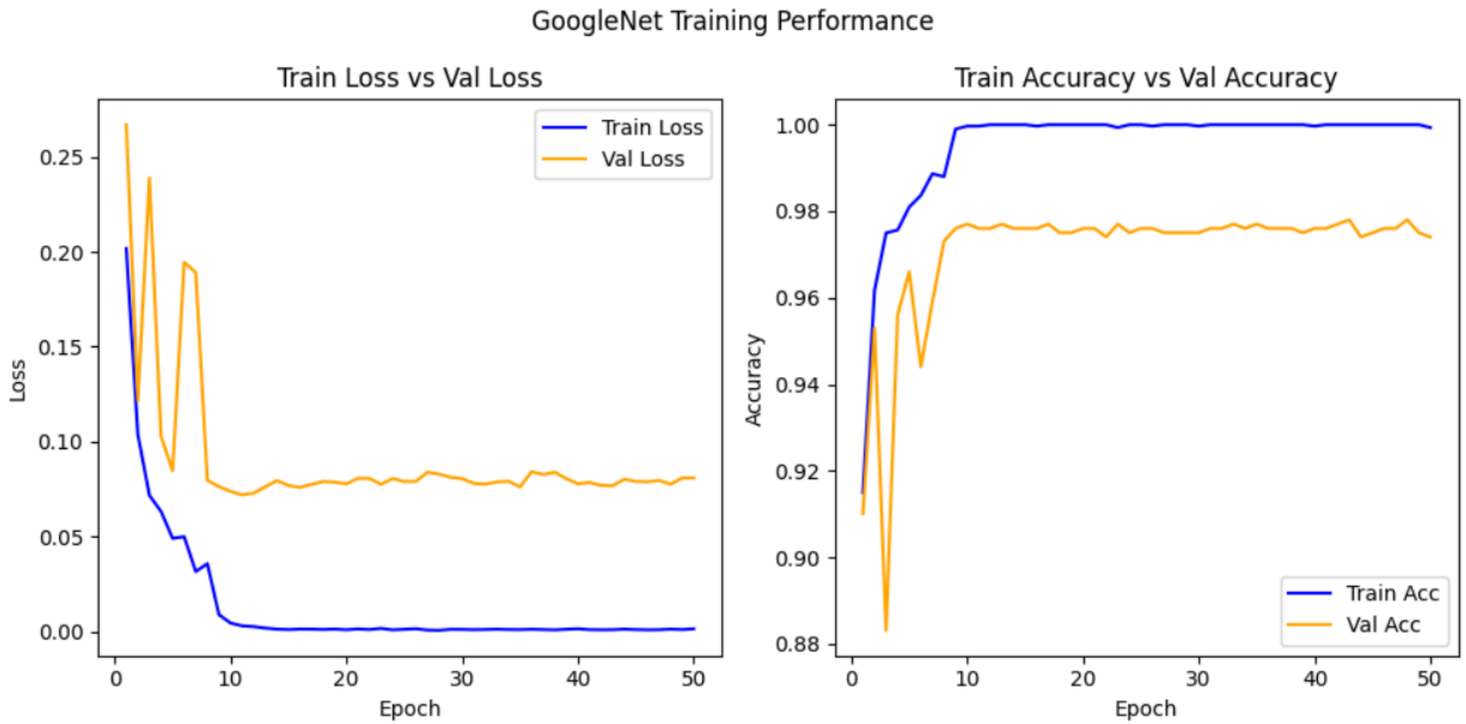
Optimizer
Adam optimizer with Lr = 0.001

FC Layer
Change the last fully connected layer into two.

Loss Function
Implementing the crossentropy loss.

Scheduler
Decrease the learning rate to optimize the model

GoogleNet Inception v1 Performance



The best validation accuracy of the model is 0.9720 with 0.0829 loss, while the training accuracy and loss are 1.000 and 0.0012. Since the difference is not significant, we conclude that this model is neither overfit nor underfit.

Model Visualization

class: female predicted: female

class: female predicted: female

class: female predicted: female

class: female predicted: female

class: female predicted: female

class: female predicted: female

class: male predicted: male

class: male predicted: male

We can see from the visualization here that all three models are able to predict the gender of the images from test dataset, i.e. the female images are predicted as female and the male images are predicted as male.

Comparisons

The results of the three models are summarized into the following table.

Model \ Phase	Train	Validation	Test	Training Duration
VGG16	Loss: 0.0089 Acc: 0.9980	Loss: 0.4487 Acc: 0.9171	Loss: 0.3606 Acc: 0.9181	48m 25s
GoogleNet	Loss: 0.0012 Acc: 1.0000	Loss: 0.0829 Acc: 0.9720	Loss: 0.0727 Acc: 0.9770	17m 44s
Resnet18	Loss: 0.0010 Acc: 1.0000	Loss: 0.0909 Acc: 0.9760	Loss: 0.0867 Acc: 0.9720	15m 10s

We obtain that the best model for this problem is GoogleNet.

Real-world Application

In real life, face recognition, especially gender classification, can be used for:

- Personalization of advertisements on smart displays (a device that can be used for face recognition) in public places such as parks, malls, and so on,
- Face recognition can be applied to smart home systems for automation settings or security,
- Face recognition can be applied to healthcare services for treatment recommendations based on gender.

Future Improvement

For future development, it may be done by:

- Training the model on a full body photo to see if it can identify gender,
- Training a model that can identify gender on moving images (gifs) or videos,
- Training the model on photos that have much noise,
- Trying other advanced models or even create a new model construction,
- Trying to use other optimizer and loss function.

Conclusion

All three experiments produce models that are neither overfit nor underfit with varying loss, accuracy, and time consumption values. However, the results of this experiment show that **the most suitable model for this case is GoogleNet,** which produces a **test loss of 0.0727** and a **test accuracy of 0.9770**. Although the **time consumption during training is 17m 44s** which is 2 minutes longer than ResNet18 at 15m 10s but it is still much faster than VGG16 at 48m 25s.

Thank You!