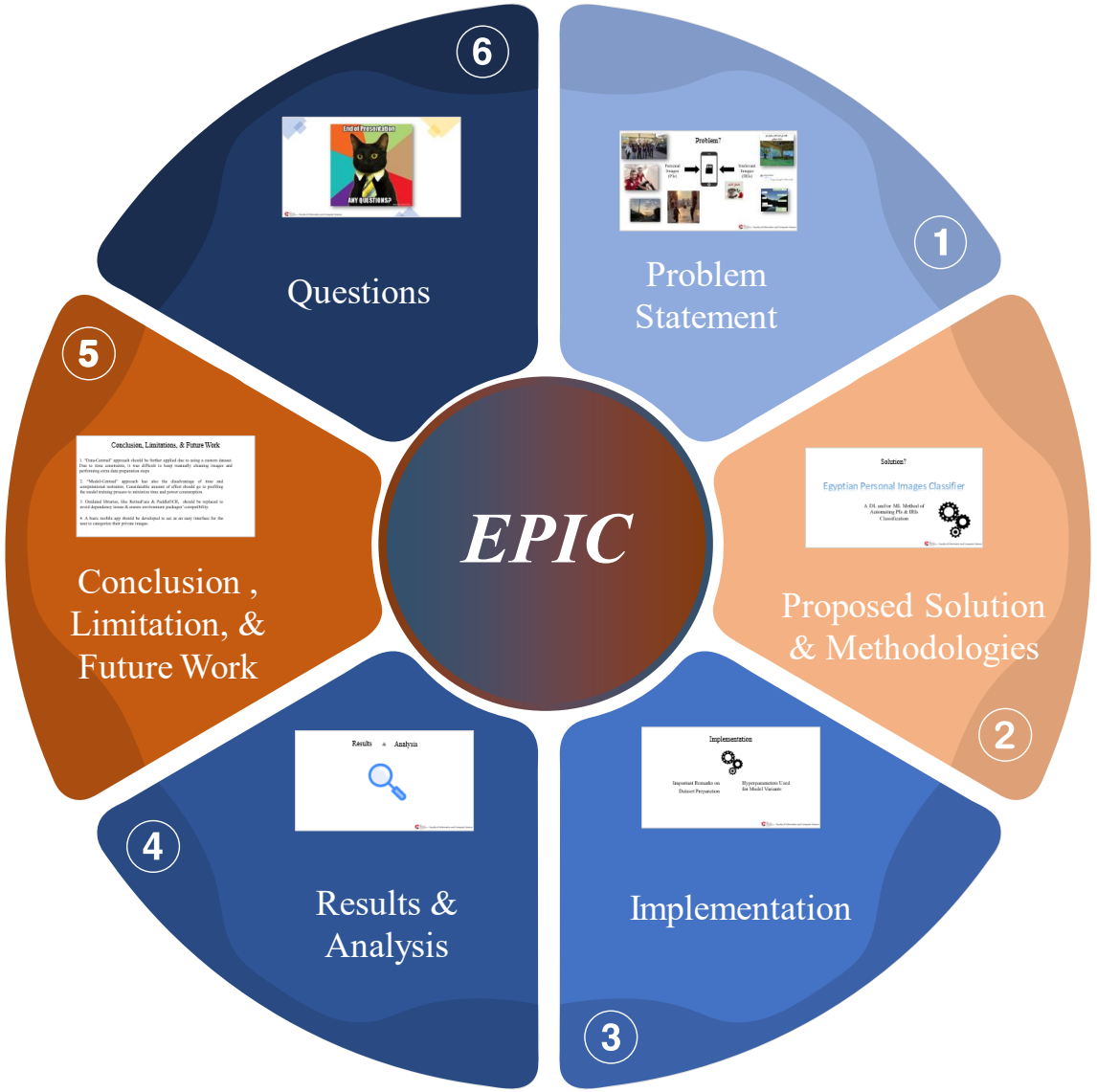
# EPIC: *Egyptian Personal Images Classifier*

By: Ashraf Adel - 196280

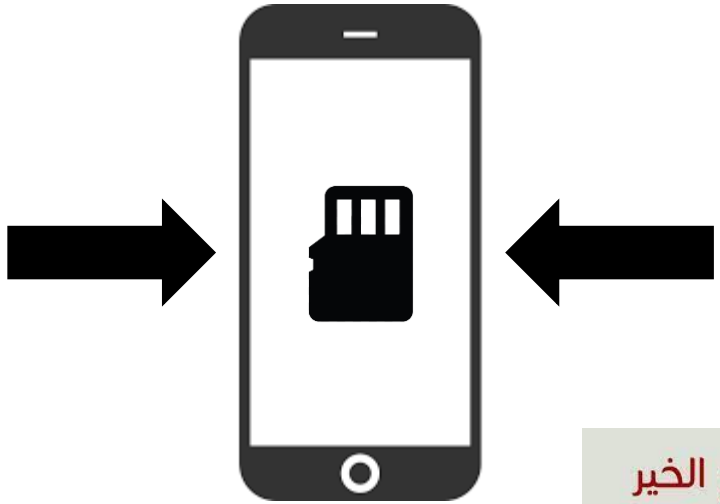
Supervised By:

Associate Prof. Nahla Barakat



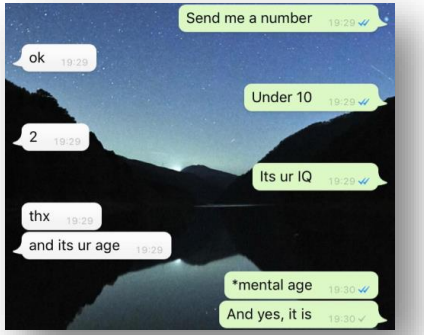
# Problem?

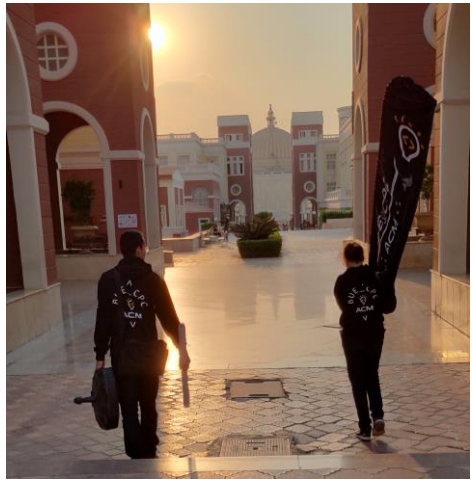
Personal Images (PIs)



Irrelevant Images (IRIs)



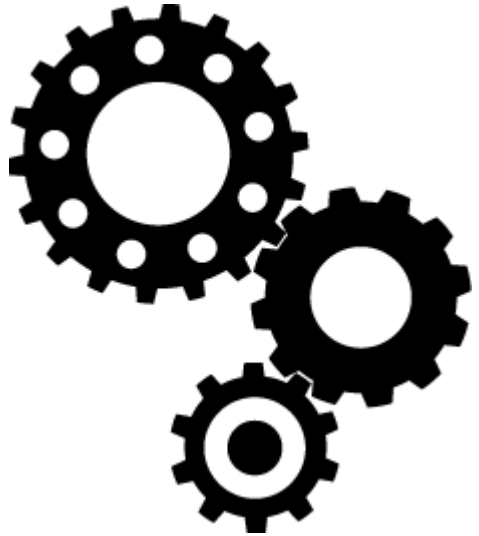



# Solution?

# Egyptian Personal Images Classifier

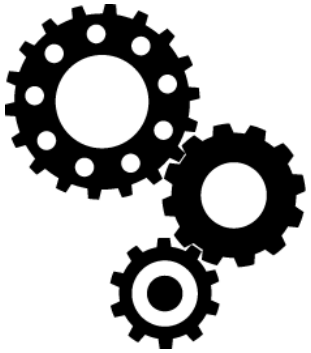
A DL and/or ML Method of
Automating PIs & IRIs
Classification



# Methodologies

1

Dataset Preparation Phase



3

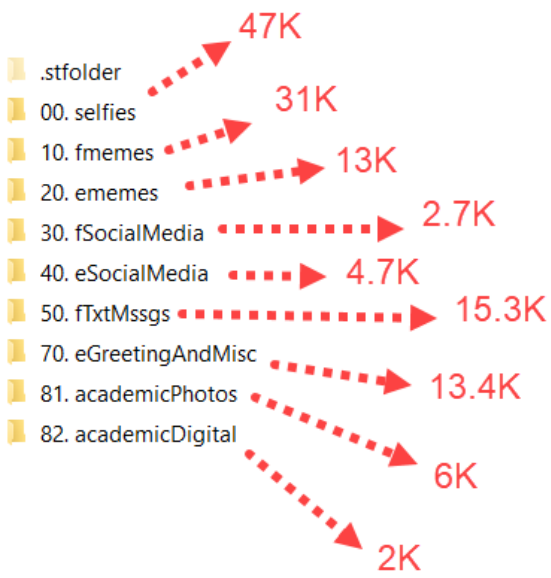
Model Evaluation Phase

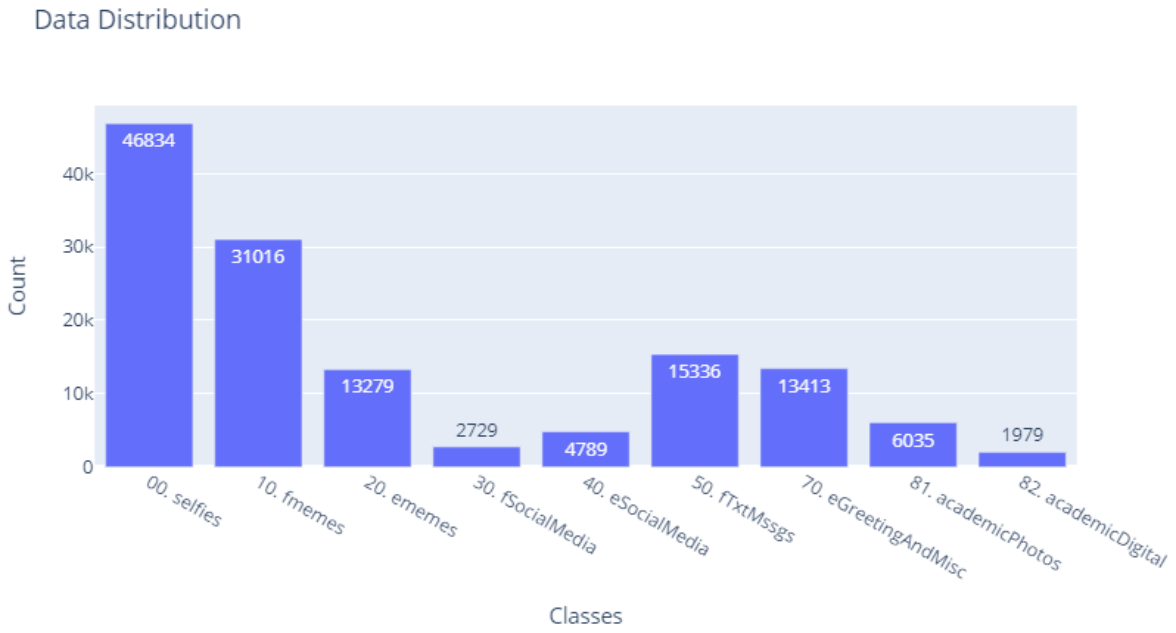
2

Model Development Phase

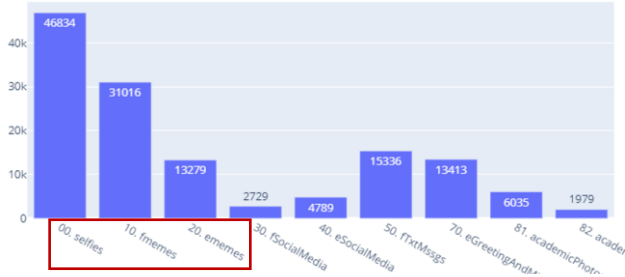
1 Dataset Preparation Phase

# Dataset Class Distribution



# 1 Dataset Preparation Phase




## Dataset Class Samples

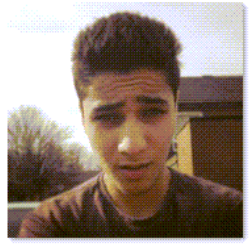
The Only Benchmark Data


selfies0001977.jpg

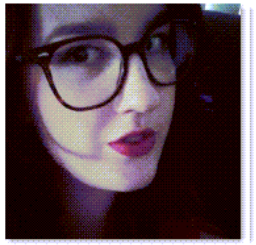

selfies0001978.jpg

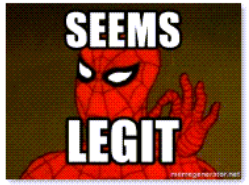

selfies0001986.jpg


selfies0001993.jpg

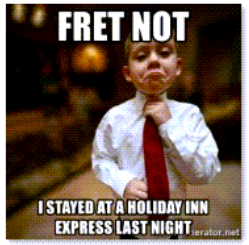

selfies0001995.jpg


selfies0001997.jpg


selfies0001999.jpg



fmemes0000000.jpg

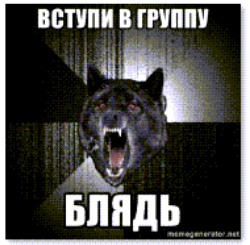


fmemes0000001.jpg

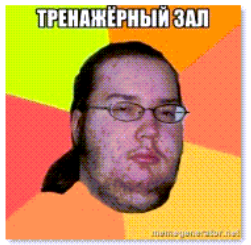
fmemes0000002.jpg

fmemes0000003.jpg

fmemes0000004.jpg



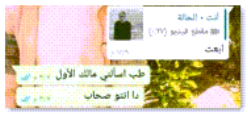
fmemes0000005.jpg



fmemes0000006.jpg

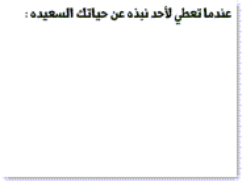
Remaining Classes are Scraped


ememes0000000.jpg

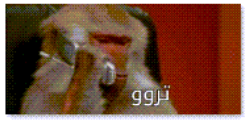

ememes0002010.jpg

ememes0002210.jpg

ememes0003015.jpg



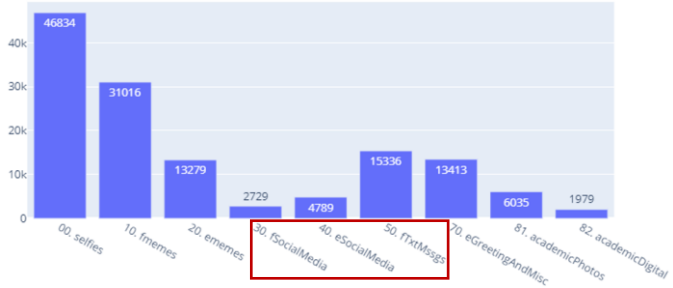
ememes0003312.jpg



ememes0005868.jpg


ememes0006874.jpg

# 1 Dataset Preparation Phase




## Dataset Class Samples

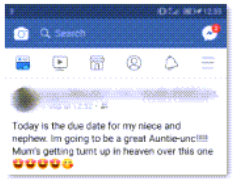
fSocialMedia0000368.jpg


fSocialMedia0000369.jpg


fSocialMedia0000370.jpg

fSocialMedia0000371.jpg

fSocialMedia0000515.jpg



fSocialMedia0000373.jpg


fSocialMedia0000375.jpg


eSocialMedia0000119.jpg

eSocialMedia0000120.jpg

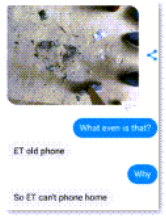
eSocialMedia0000121.jpg

eSocialMedia0000122.jpg


eSocialMedia0000123.jpg

eSocialMedia0000124.jpg

eSocialMedia0000458.jpg

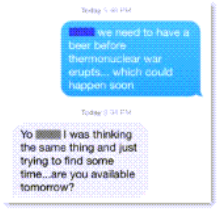


fTxtMssgs0000162.jpg


fTxtMssgs0000163.jpg


fTxtMssgs0000164.jpg


fTxtMssgs0000165.jpg



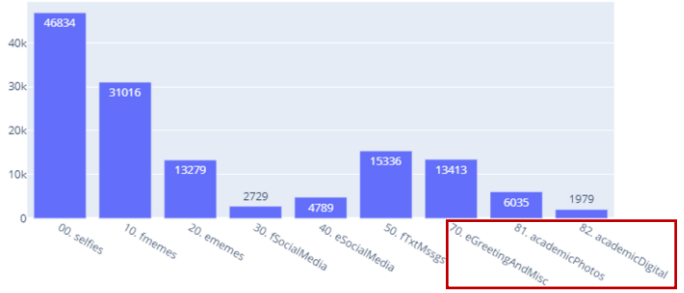
fTxtMssgs0000166.jpg

fTxtMssgs0000167.jpg

fTxtMssgs0000168.jpg

# 1 Dataset Preparation Phase

## Dataset Class Samples

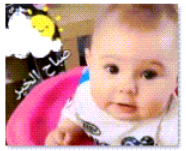




eGreetingAndMisc0000000.jpg


eGreetingAndMisc0000001.jpg

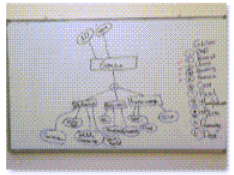

eGreetingAndMisc0000002.jpg

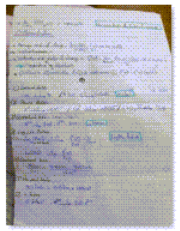

eGreetingAndMisc0000023.jpg

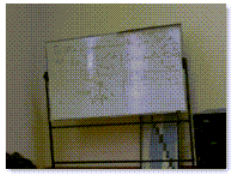

eGreetingAndMisc0000156.jpg

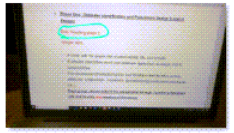
eGreetingAndMisc0000005.jpg

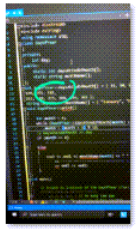
eGreetingAndMisc0005467.jpg

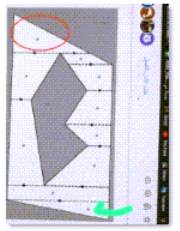

academicPhotos0000472.jpg


academicPhotos0000555.jpg


academicPhotos0001599.jpg


academicPhotos0003489.jpg

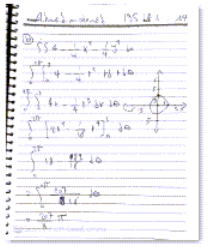

academicPhotos0004161.jpg

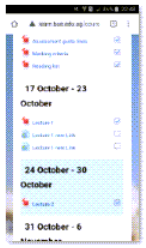

academicPhotos0005205.jpg


academicPhotos0006020.jpg

academicDigital0001857.jpg

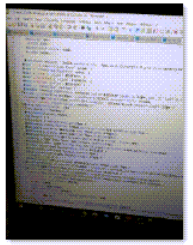

academicDigital0001858.jpg

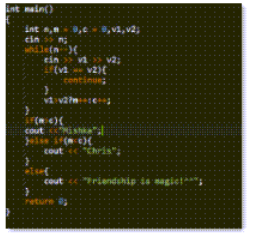

academicDigital0001722.jpg



academicDigital0000689.jpg

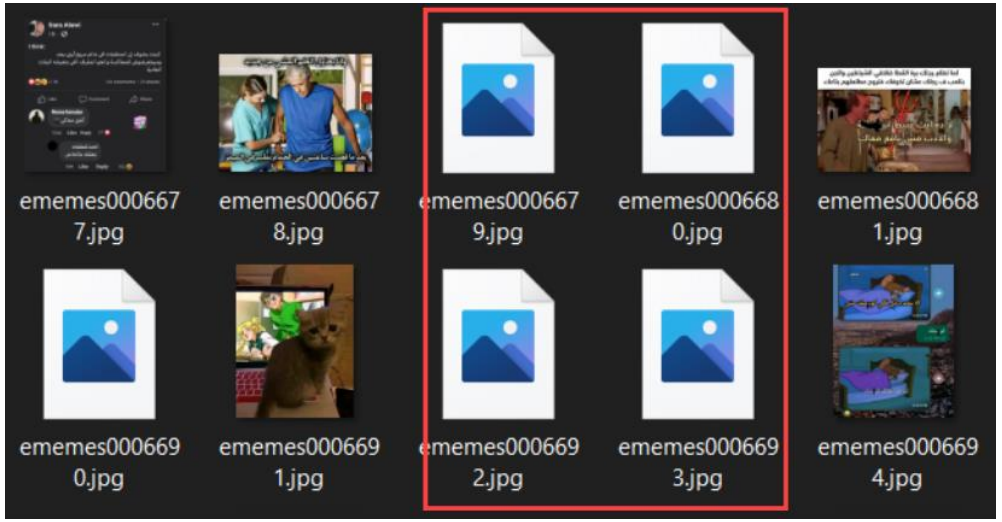

academicDigital0000253.jpg


academicDigital0000004.jpg

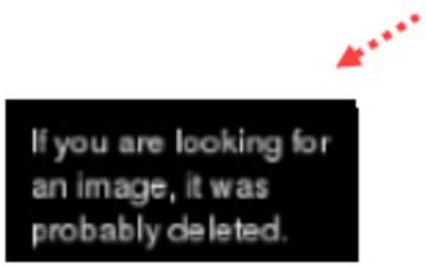

academicDigital0000188.jpg

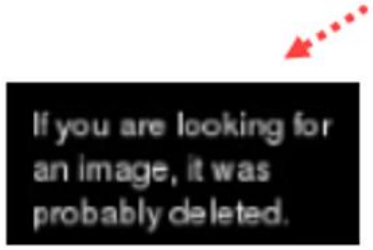
# Dataset Cleaning

## Removing Corrupt Images




## Removing Images Downloaded from Dead Links



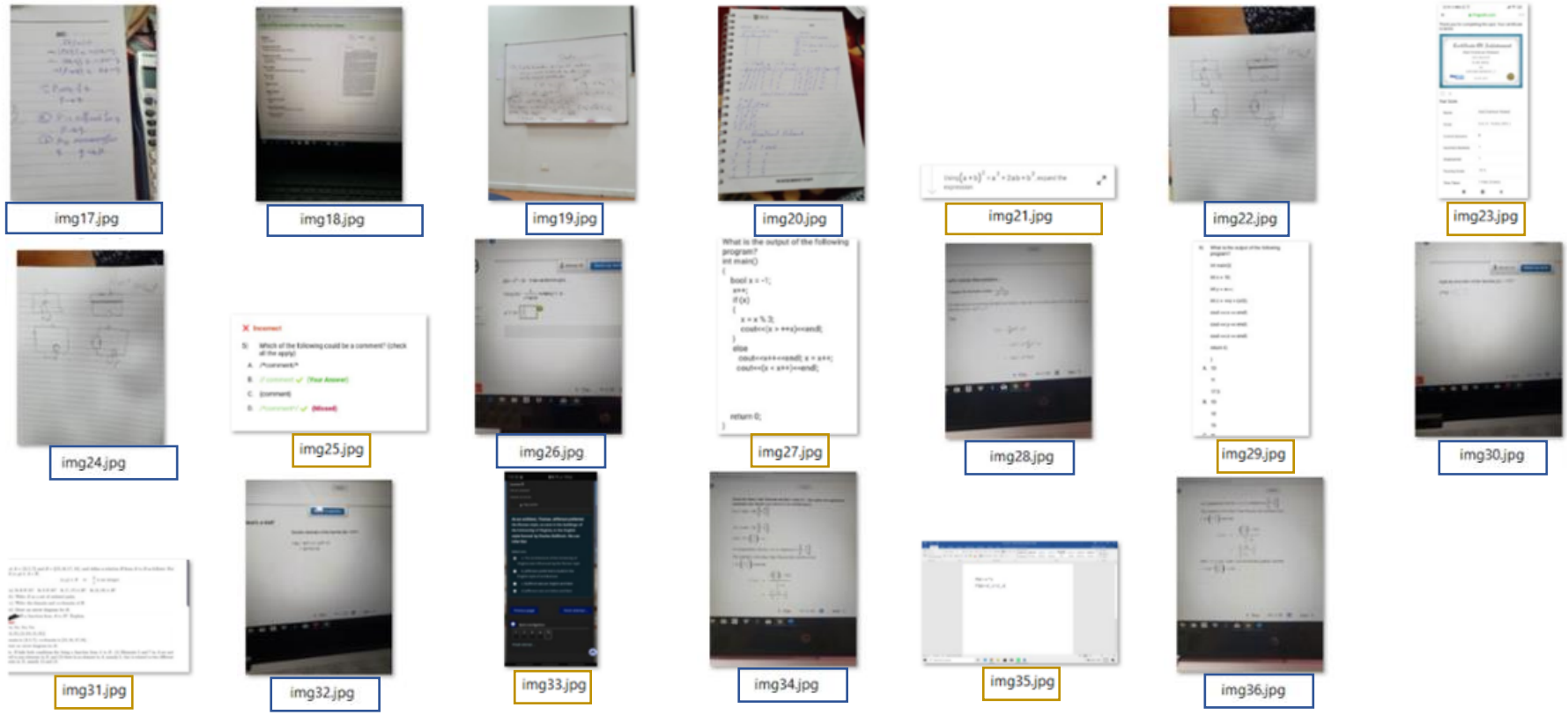
# 1 Dataset Preparation Phase

# Dataset Cleaning

"academicPhotos" (contains "academicDigital" which should be removed)

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names

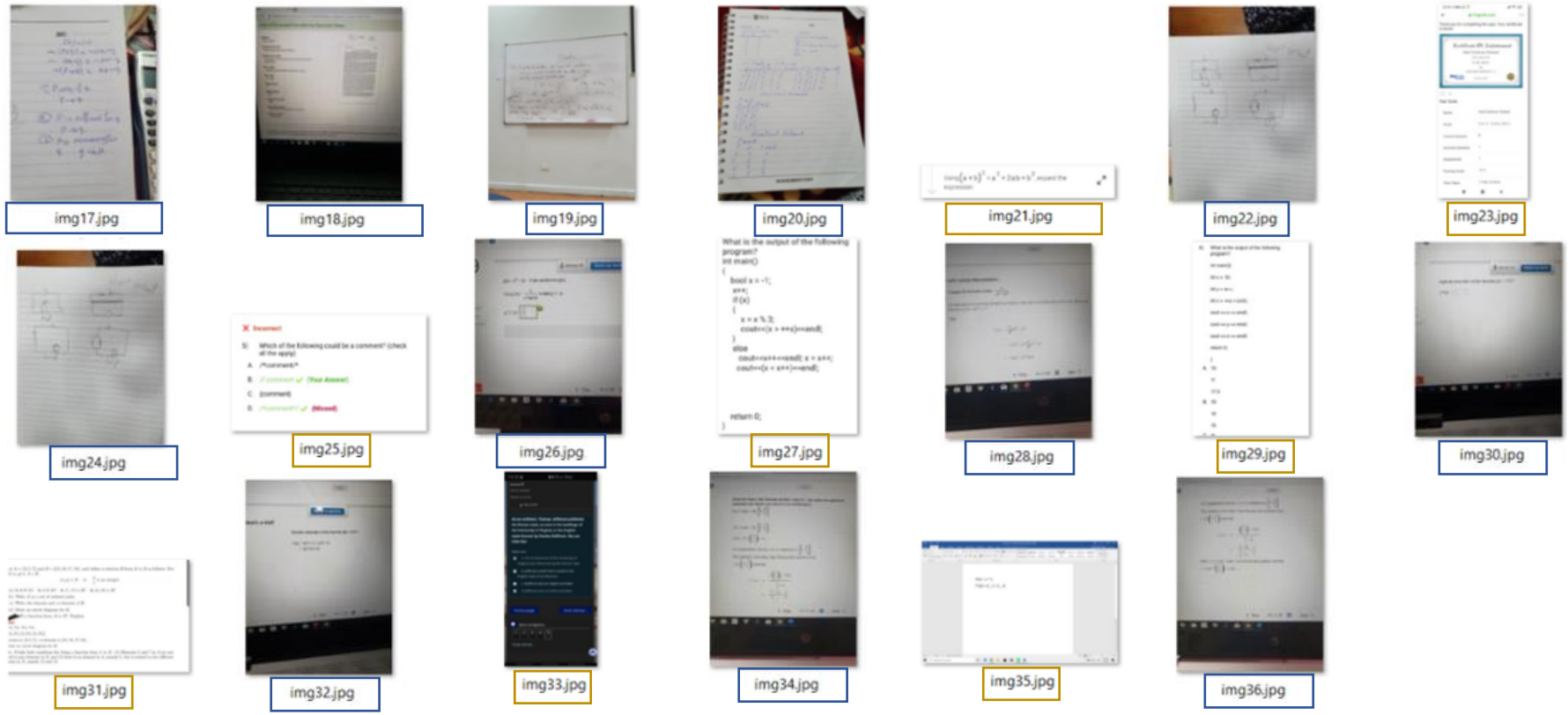


# 1 Dataset Preparation Phase

# Dataset Cleaning

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image — based on metrics like highest ratio, #text boxes, etc
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names

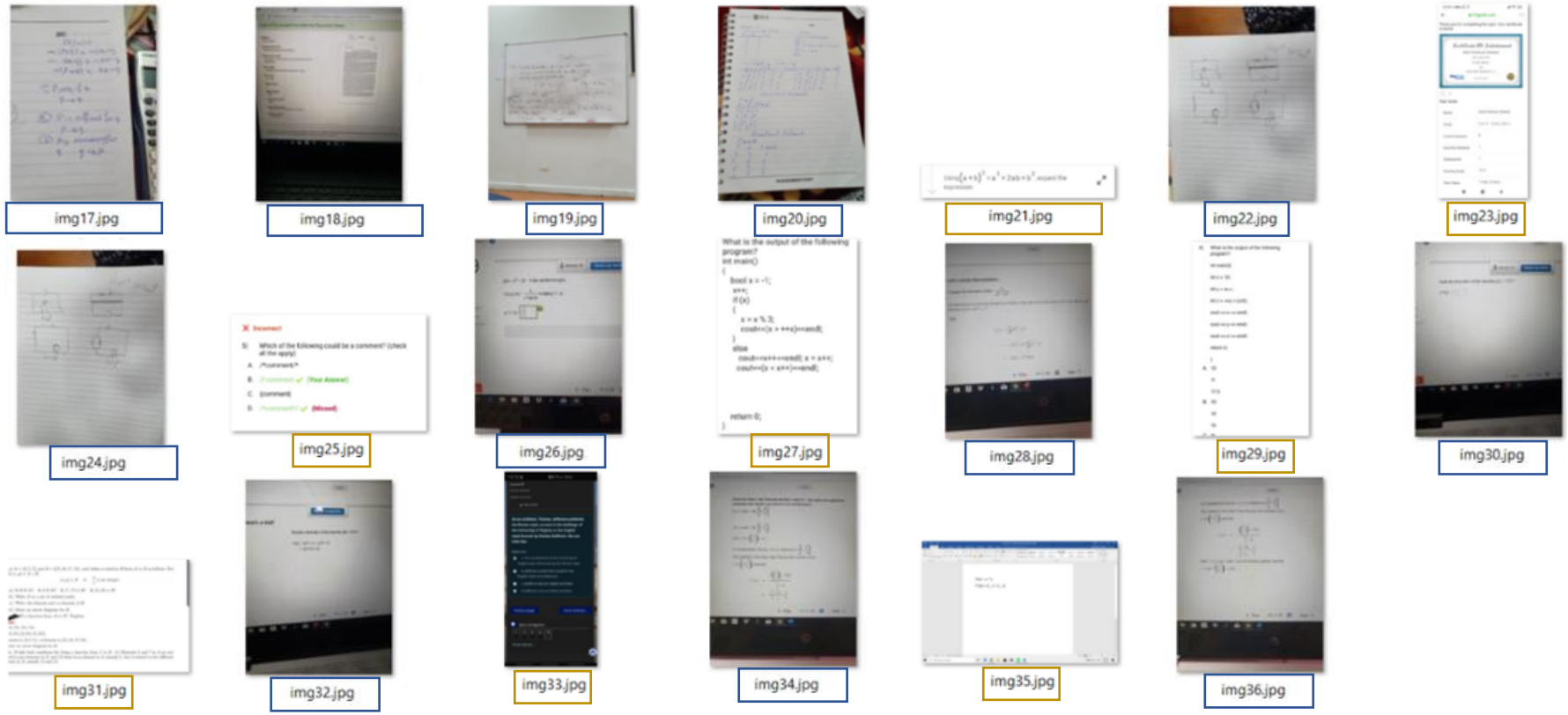


# 1 Dataset Preparation Phase

# Dataset Cleaning

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names



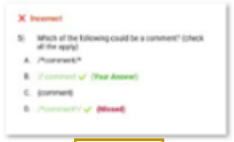
# 1 Dataset Preparation Phase

# Dataset Cleaning

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names


img29.jpg


img25.jpg

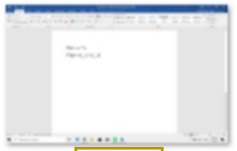

img27.jpg


img21.jpg


img23.jpg

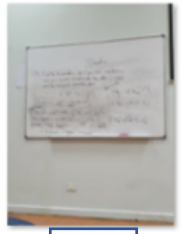

img31.jpg


img33.jpg


img35.jpg


img17.jpg


img18.jpg


img19.jpg


img20.jpg


img22.jpg


img24.jpg


img26.jpg


img28.jpg


img30.jpg


img32.jpg


img34.jpg


img36.jpg

# 1 Dataset Preparation Phase

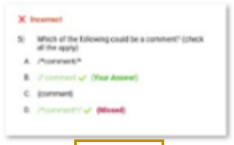
# Dataset Cleaning

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names

0000_img29


img29.jpg


img25.jpg


img27.jpg

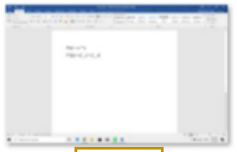

img21.jpg


img23.jpg


img31.jpg


img33.jpg

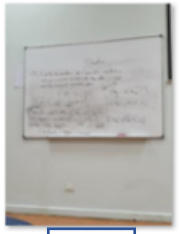
0007_img35


img35.jpg

0008_img17


img17.jpg


img18.jpg


img19.jpg


img20.jpg


img22.jpg


img24.jpg


img26.jpg


img28.jpg


img30.jpg


img32.jpg


img34.jpg


img36.jpg

0028_img36

# 1 Dataset Preparation Phase

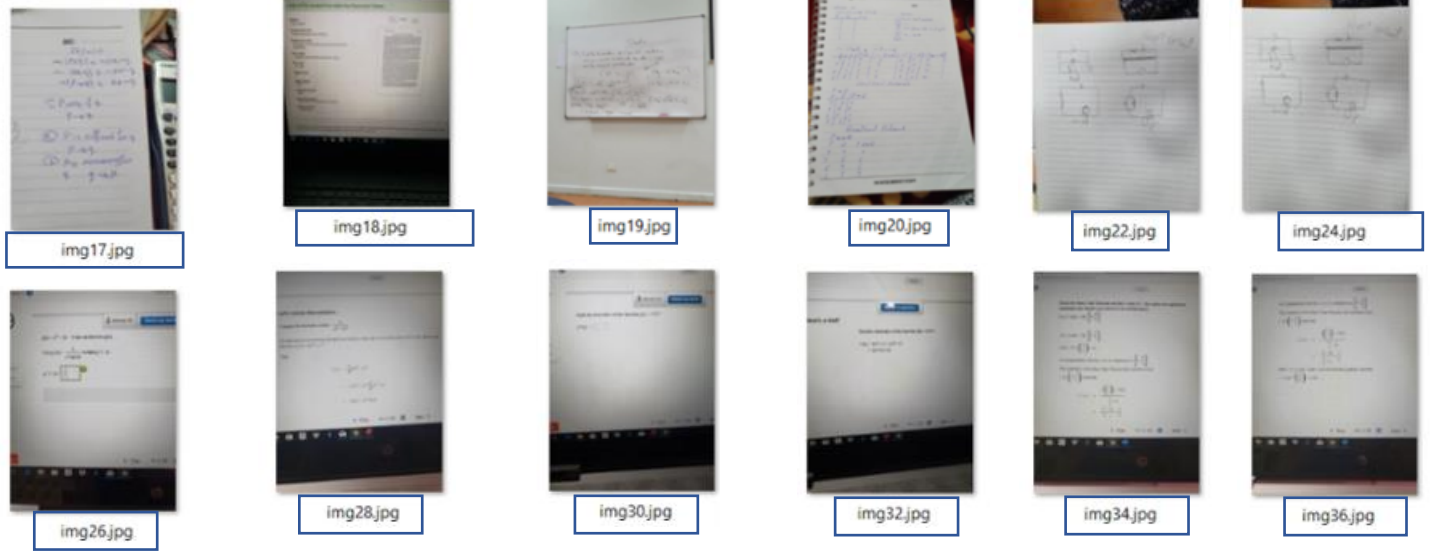
# Dataset Cleaning

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names

move them to another folder or delete them

0008_img17




0028_img36

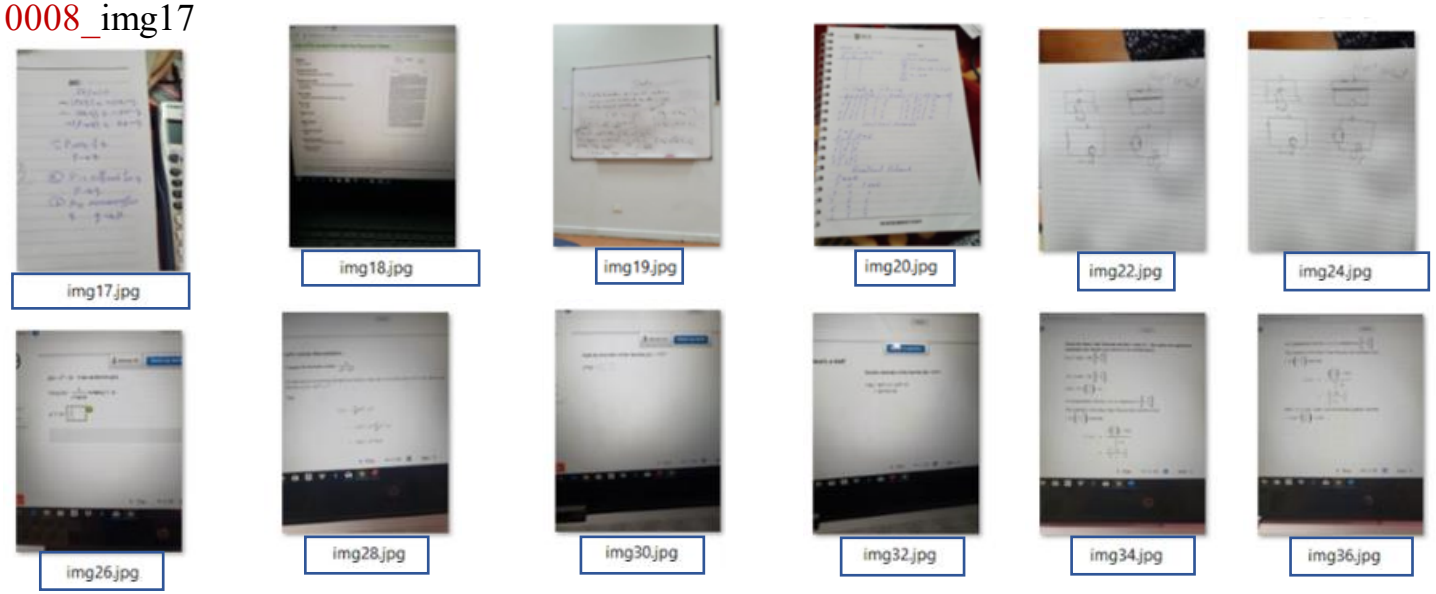
# Dataset Cleaning

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names

move them to another folder or delete them

0008_img17



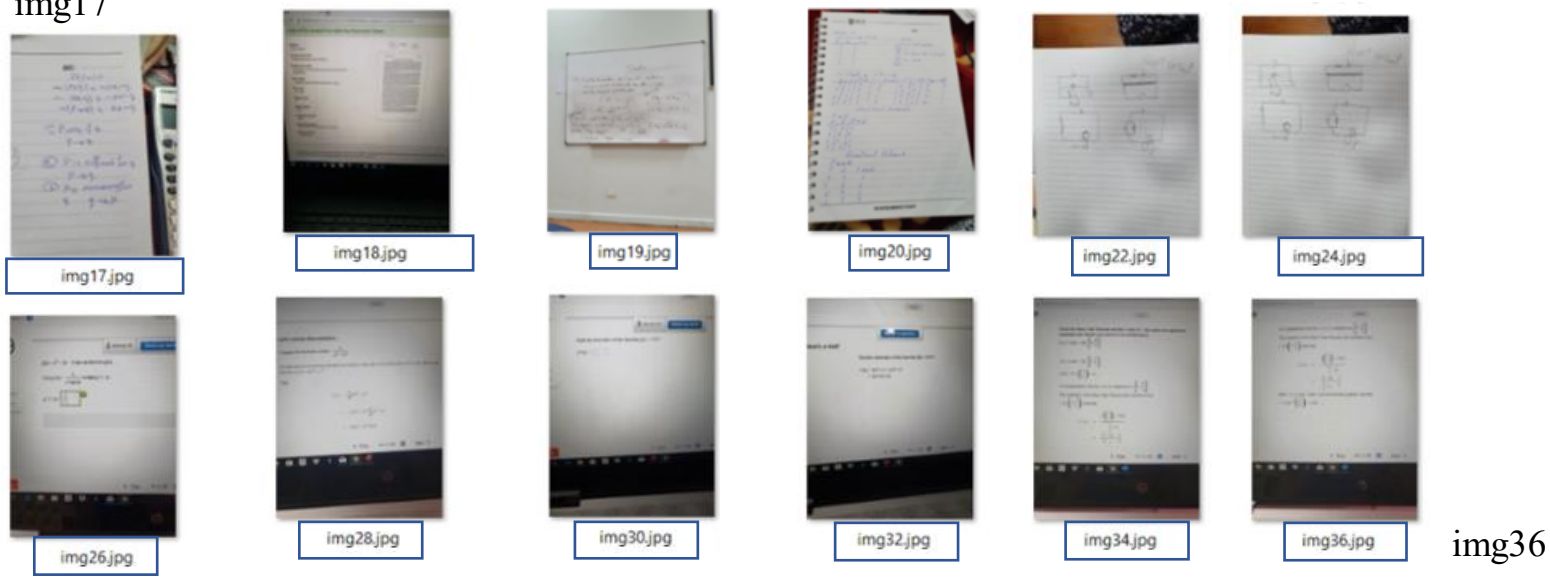

0028_img36

# Dataset Cleaning

Pipeline for Removing Misclassified/Outlier labels from a Chosen Class:

1. Choose class
2. Assign value for each image
3. Sort images based on their values
4. Rename sorted images in class directory
5. Manually remove unwanted images
6. Rename images to original names

img17




img36

# Dataset Cleaning

## Removing Outlier Clusters

(Generated from KNN with #clusters = 20)

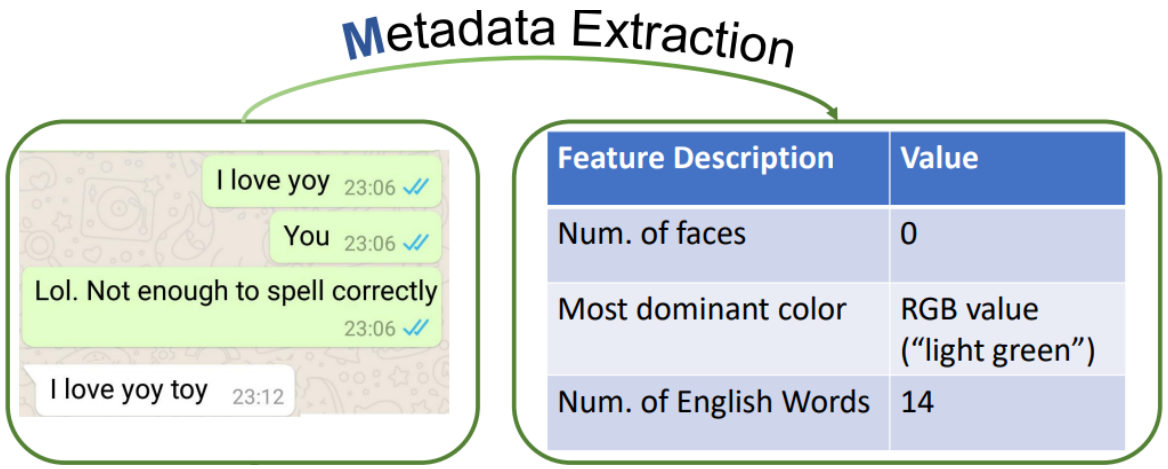
fastdup by Visual Layer

| Info | |
|---|---|
| component | 6 |
| num_images | 48 |
| mean_distance | 106.078857 |

Example: “academicDigital” images which should actually be in “academicPhotos”

# Dataset Preprocessing

Image's Metadata Example:

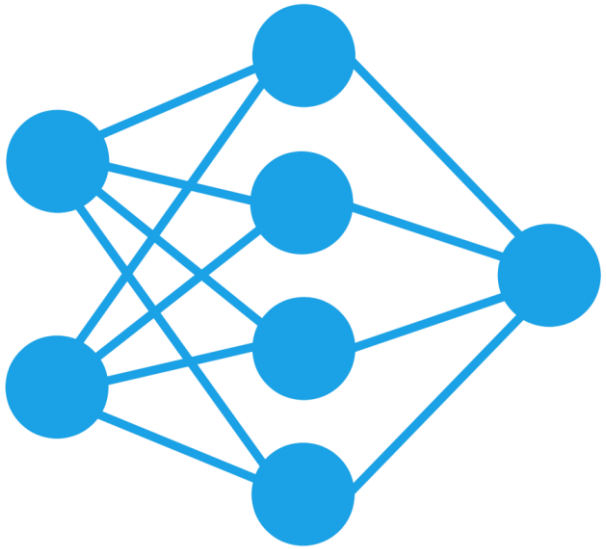



| Feature Description | Value |
|---|---|
| Num. of faces | 0 |
| Most dominant color | RGB value ("light green") |
| Num. of English Words | 14 |

List of Features (i.e., Metadata) Extracted per Image

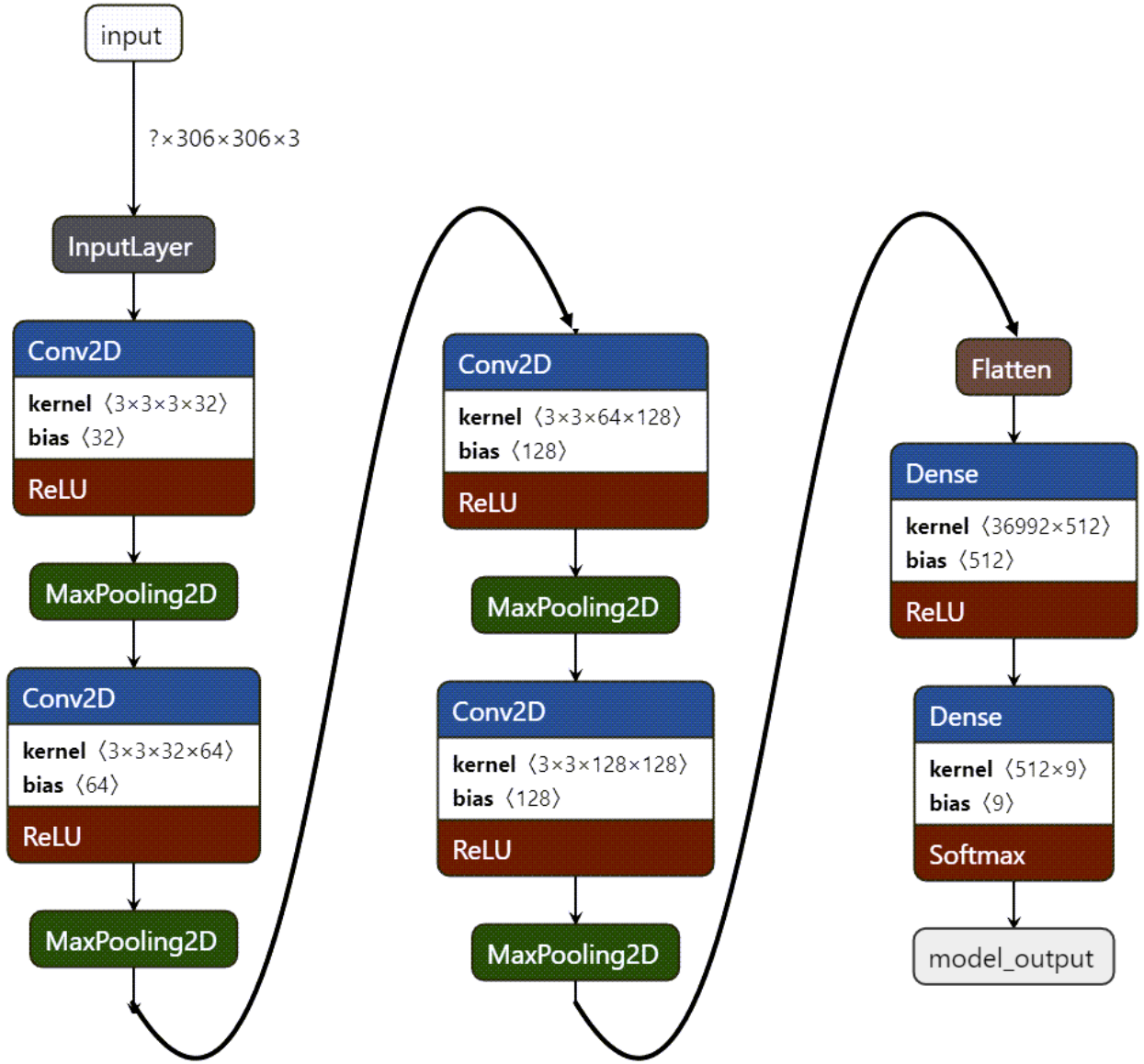
```
'file_name', 'relative_path', 'aspect_ratio', 'area', 'width', 'height',
'dominant_color_1', 'color_to_image_ratio_1', 'dominant_color_2',
'color_to_image_ratio_2', 'dominant_color_3', 'color_to_image_ratio_3',
'dominant_color_4', 'color_to_image_ratio_4', 'dominant_color_5',
'color_to_image_ratio_5', 'total_faces_count', 'faces_to_image_ratio',
'text_to_image_ratio', 'lines_per_text_block',
'total_lines_of_text_count', 'text_blocks_count', 'total_en_boxes',
'total_ar_boxes', 'en_words_original', 'ar_words_original',
'en_words_lemmatized', 'en_words_segmented_and_lemmatized',
'ar_words_lemmatized', 'en_words_including_separators',
'ar_words_including_separators', 'en_words_to_boxes',
'ar_words_to_boxes', 'en_avg_score', 'ar_avg_score',
'math_digits_or_symbols_count', 'class'
```


# Model Development Phase



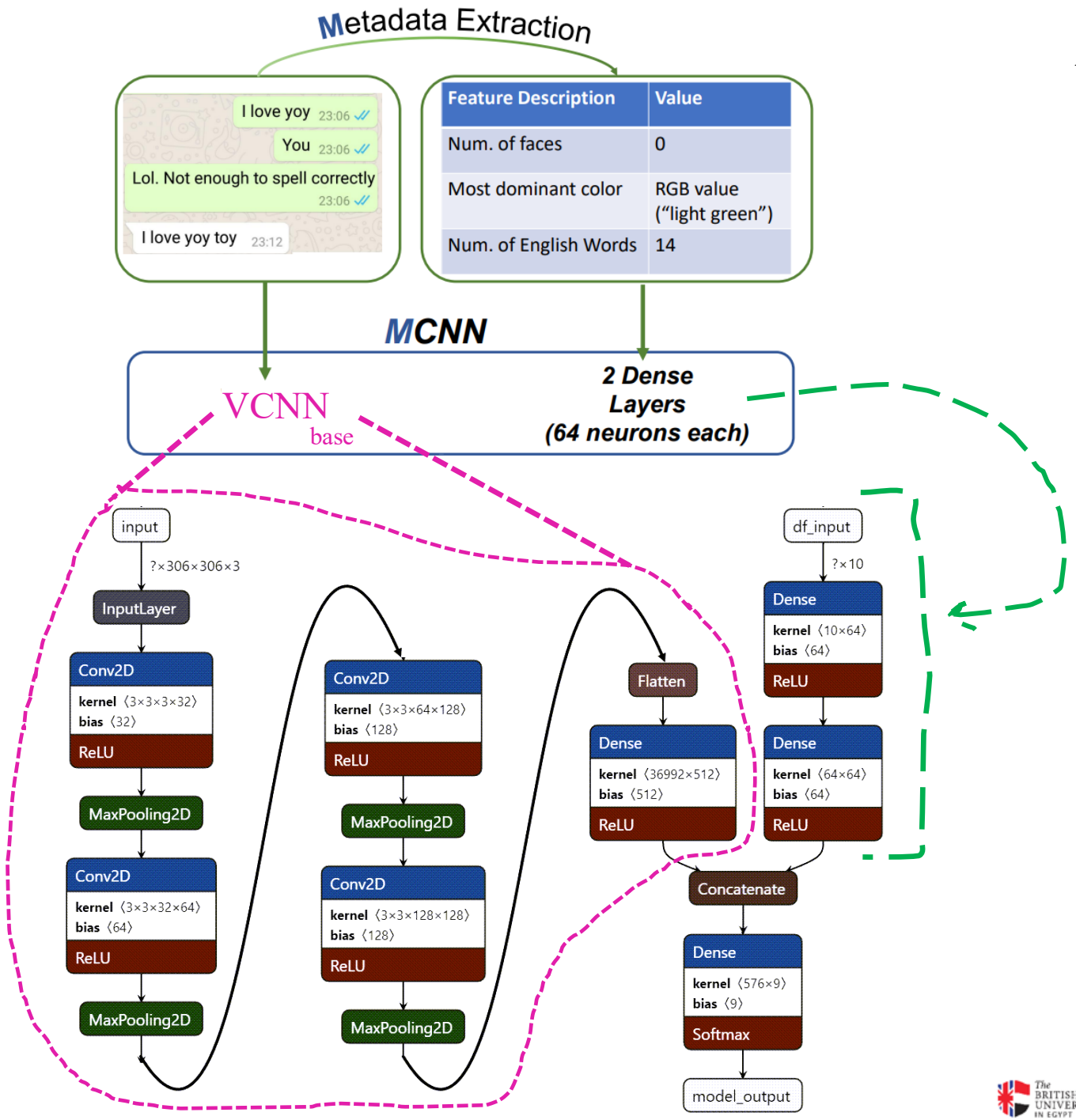
$VCNN_{base}$

“m01”



# MCNN

"m02"


Metadata Extraction
I love yoy 23:06
You 23:06
Lol. Not enough to spell correctly 23:06
I love yoy toy 23:12
Feature Description
Value
Num. of faces
0
Most dominant color
RGB value ("light green")
Num. of English Words
14
MCNN
VCNN
base
2 Dense Layers (64 neurons each)
input
?×306×306×3
InputLayer
Conv2D
kernel ⟨3×3×3×32⟩
bias ⟨32⟩
ReLU
MaxPooling2D
Conv2D
kernel ⟨3×3×32×64⟩
bias ⟨64⟩
ReLU
MaxPooling2D
Conv2D
kernel ⟨3×3×64×128⟩
bias ⟨128⟩
ReLU
MaxPooling2D
Conv2D
kernel ⟨3×3×128×128⟩
bias ⟨128⟩
ReLU
MaxPooling2D
Flatten
Dense
kernel ⟨36992×512⟩
bias ⟨512⟩
ReLU
df_input
?×10
Dense
kernel ⟨10×64⟩
bias ⟨64⟩
ReLU
Dense
kernel ⟨64×64⟩
bias ⟨64⟩
ReLU
Concatenate
Dense
kernel ⟨576×9⟩
bias ⟨9⟩
Softmax
model_output

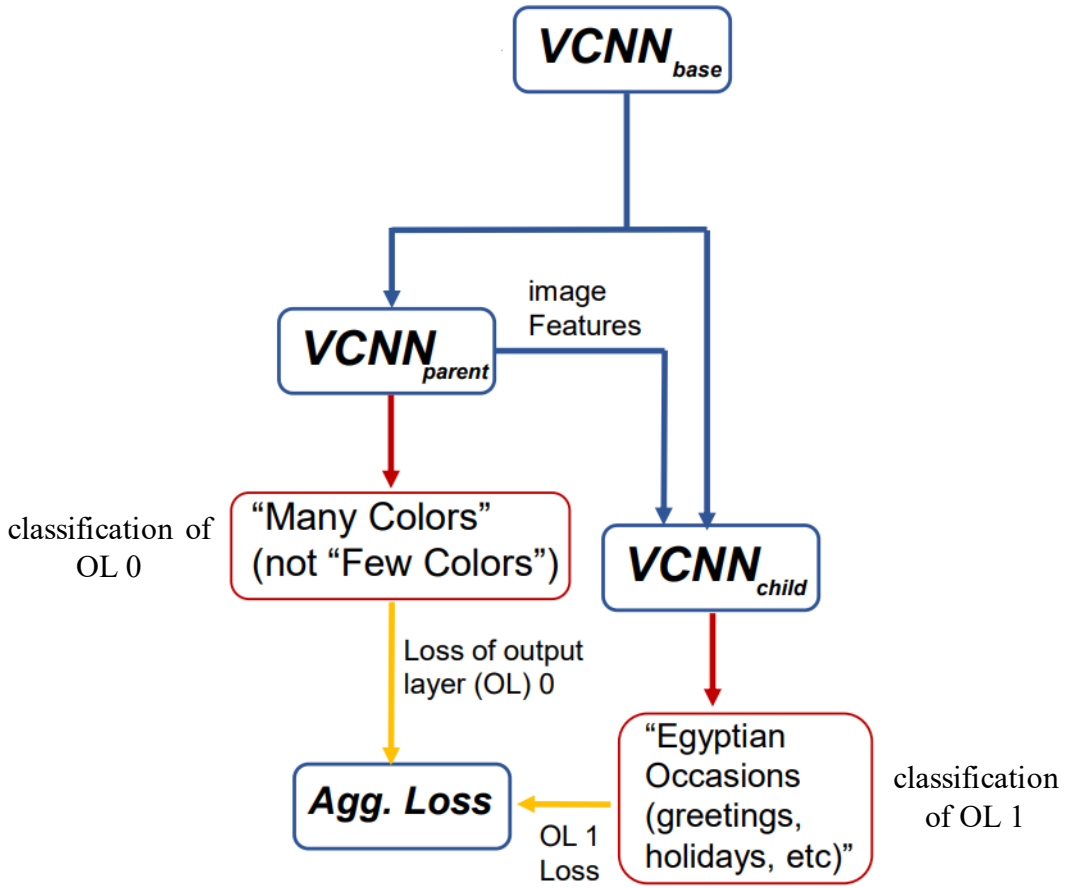
## HCNN

"m03"

Output Layer 0:
Many or Few Colors

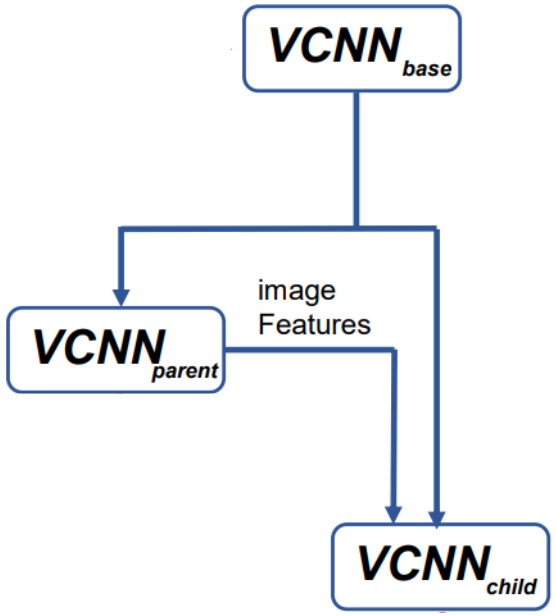
Output Layer 1:
Flat classes
(i.e., the 9 classes of the dataset)

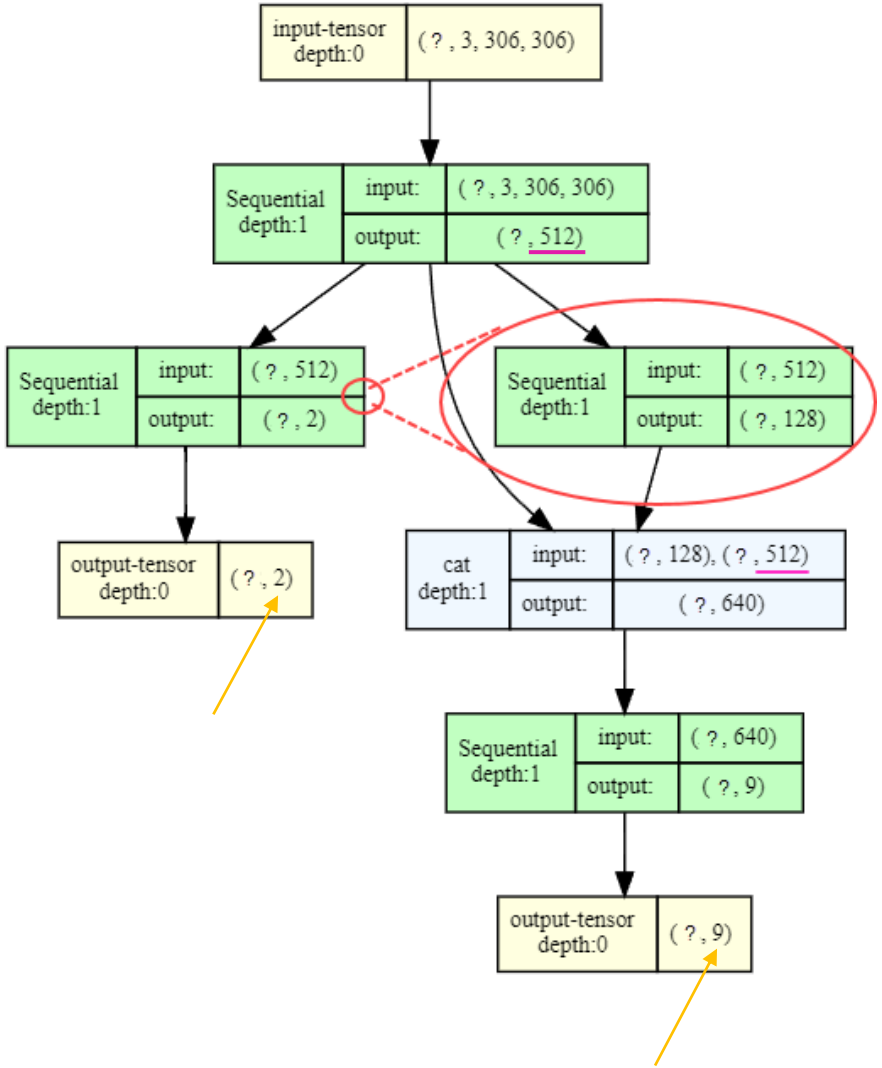
```
Input Image
└── Colors
    ├── Many Colors
    │   ├── Selfies
    │   ├── fMemes
    │   ├── eMemes
    │   └── eGreetingAndMisc
    └── Few Colors
        ├── fSocialMedia
        ├── eSocialMedia
        ├── fTxtMssgs
        ├── academicPhotos
        └── academicDigital
```



## HCNN

"m03"

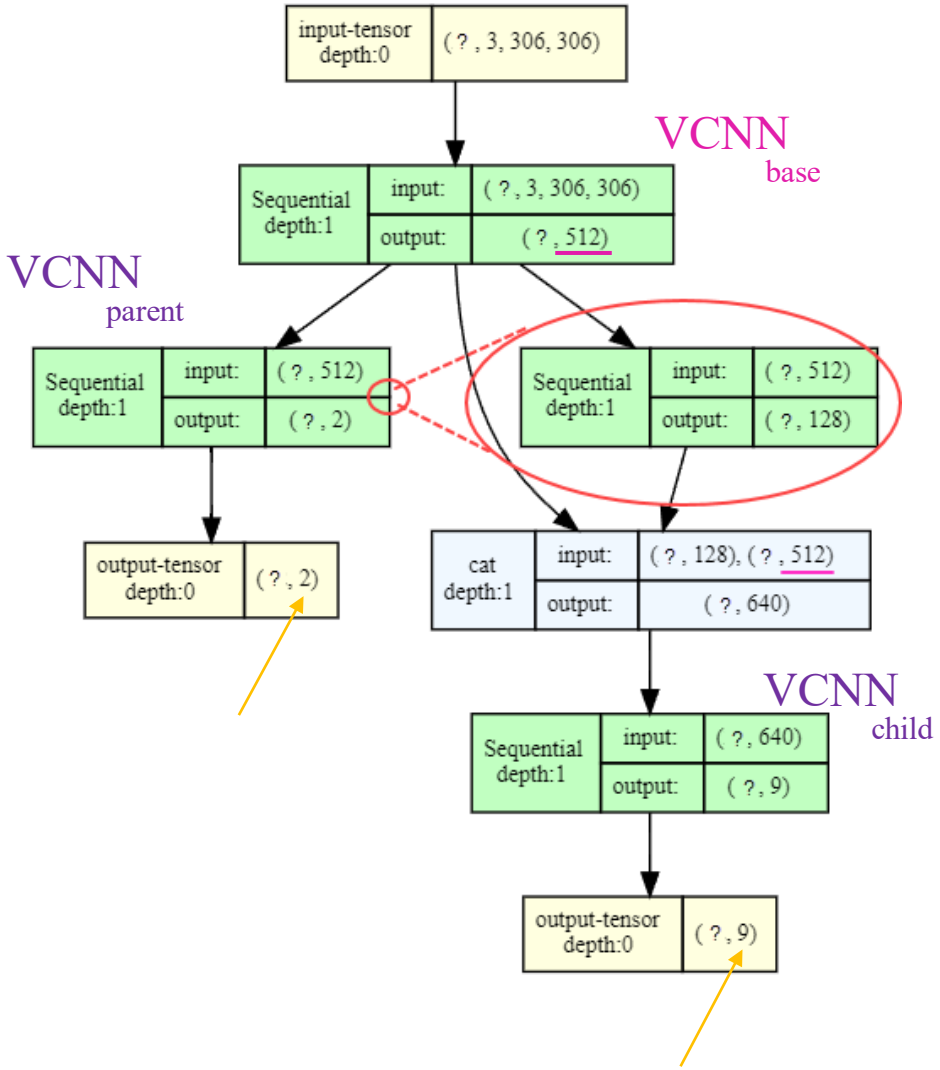

VCNN base
VCNN parent
image Features
VCNN child


input-tensor depth:0
( ? , 3, 306, 306)
Sequential depth:1
input:
( ? , 3, 306, 306)
output:
( ? , 512)
Sequential depth:1
input:
( ? , 512)
output:
( ? , 2)
Sequential depth:1
input:
( ? , 512)
output:
( ? , 128)
output-tensor depth:0
( ? , 2)
cat depth:1
input:
( ? , 128), ( ? , 512)
output:
( ? , 640)
Sequential depth:1
input:
( ? , 640)
output:
( ? , 9)
output-tensor depth:0
( ? , 9)

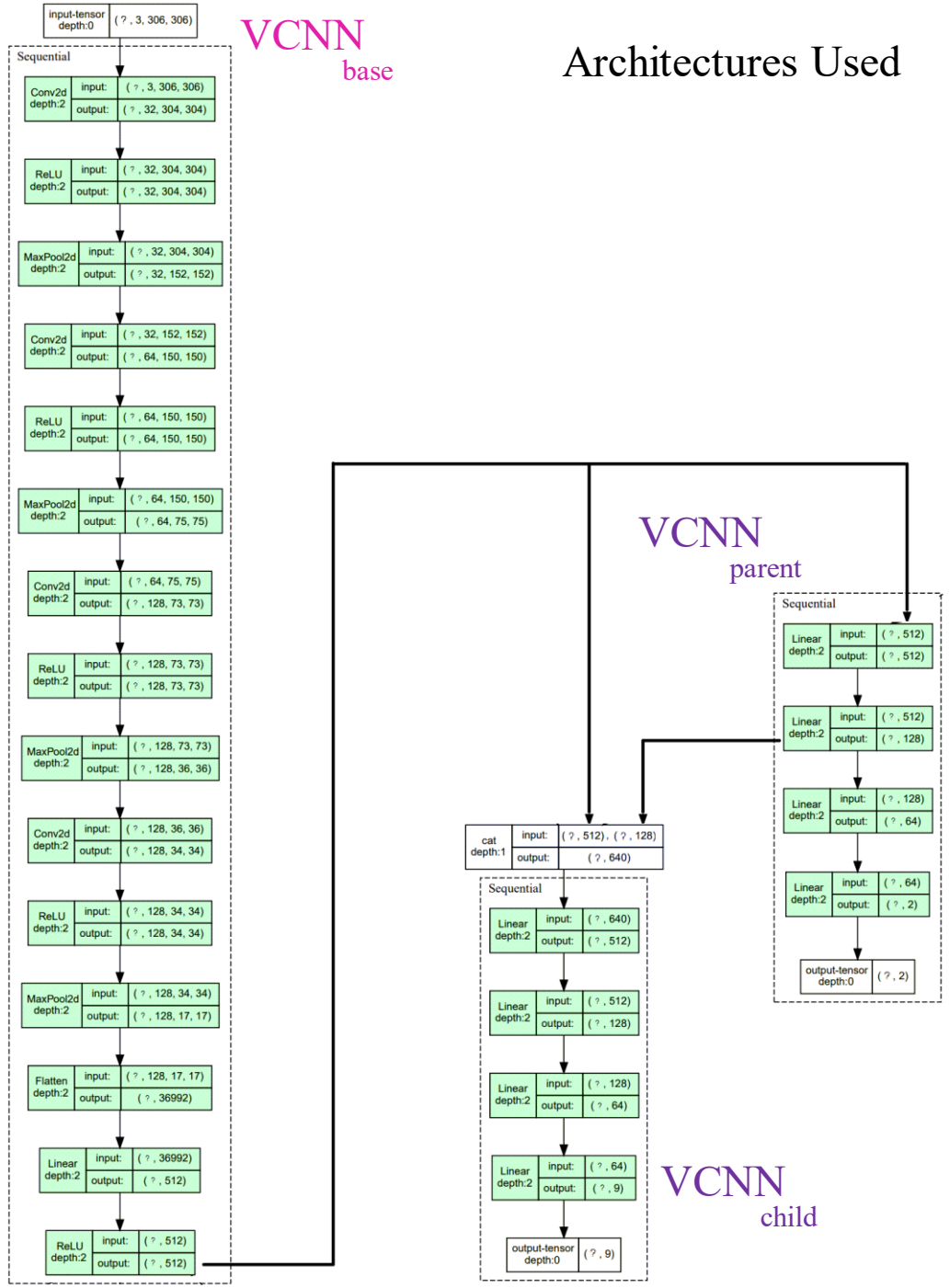
# 2 Model Development Phase

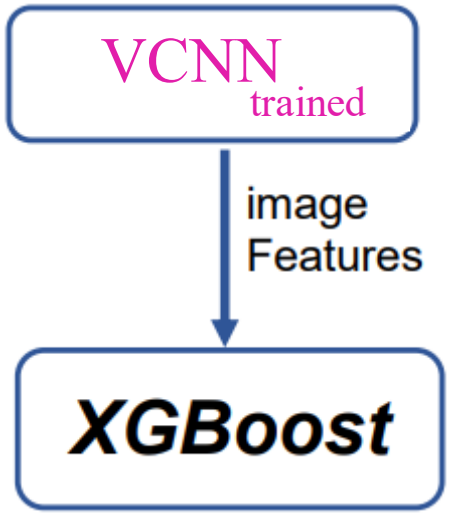
## HCNN

"m03"




## Architectures Used






XCNN

“m04”

# Objective Functions

Used on VCNN, MCNN, HCN

The **cross-entropy** cost of a $K$-class network

penalty increases when misclassifying with high probability

$$C_{\text{CE}} = -\frac{1}{n}\sum_{x}\sum_{k=1}^{K}(y_k \ln a_k^L + (1-y_k)\ln(1-a_k^L))$$

where $x$ is an input and $n$ is the number of examples in the input set.

Note that, for each $x$, only one of the $y_k$ is 1 and the rest are 0 (i.e. one-hot encoding).

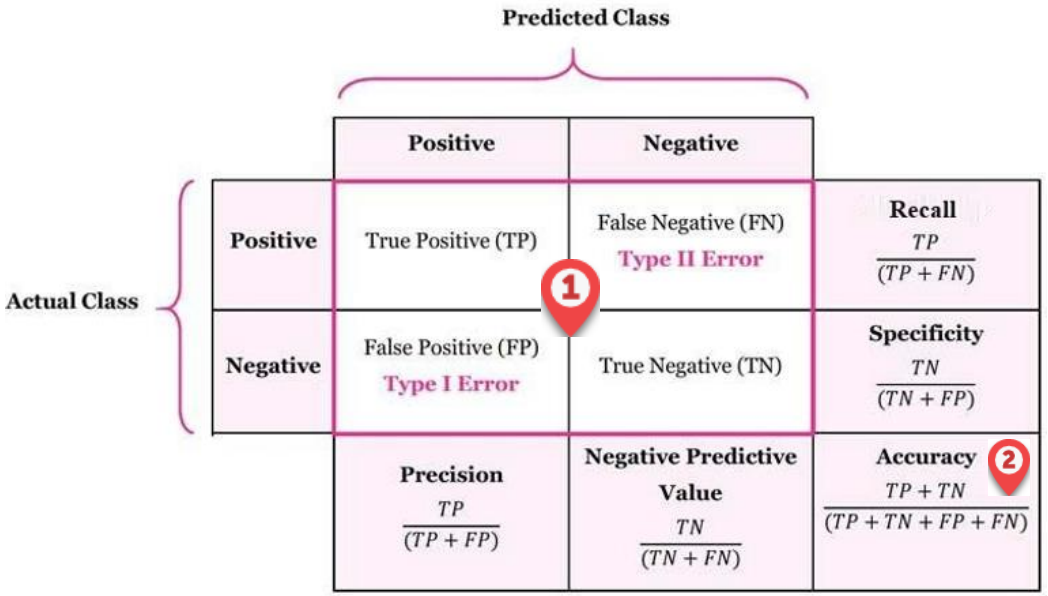
$y$ is the (one-hot encoded) desired output and $a^L$ is the output of the model.

Used on XCNN

The **log-likelihood** cost of a $K$-class network is

$$C_{\text{LL}} = -\frac{1}{n}\sum_{x} y^T \ln(a^L) = -\frac{1}{n}\sum_{x}\sum_{k=1}^{K} y_k \ln(a_k^L)$$

# Evaluation Metrics




$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$ 3

where:

$$precision = \frac{TP}{TP + FP}$$

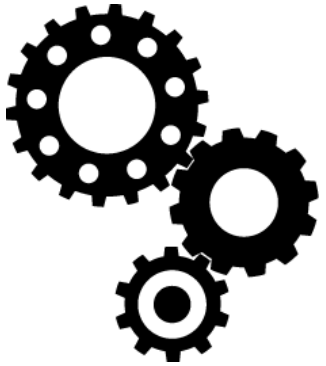
$$recall = \frac{TP}{TP + FN}$$

In "macro" F1 a separate F1 score is calculated for each value and then averaged. 4

However, only 3 and 4 are mentioned in Results Section due to their Importance on Imbalanced Datasets

up next: Implementation

# Implementation



Important Remarks on Dataset Preparation

Hyperparameters Used for Model Variants

# Dataset Preparation

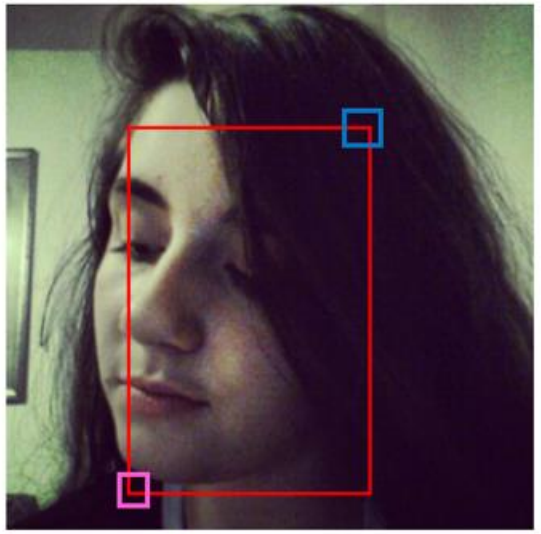
(extracting face and text metadata)

```
detections = RetinaFace.detect_faces(img)
        # rest of code...
print(total_faces, face_area_ratio)
display(img)
```

top right
x2, y2

```
{'face_1': {'score': 0.9973350167274475, 'facial_area': [70, 71, 211, 285],
1 0.3222479
```

bottom left
x1, y1



"RetinaFace" output


"PaddleOCR" output

# Hyperparameters & Descriptions of Model Variants

| Model Type | Model Version | Description |
|---|---|---|
| VCNN | 01 | no metadata, 8 batch size, 30 epochs, no augmentation. |
| | 01.1 | Same as m01, but with 32 batch size |
| | 01.3 | Same as m01, but with 16 batch size |
| | 01.4 | Same as m01.3, but with augmentation by rotating images |
| | 01.6 | Same as m01.1, but on dataset "v01.1" instead of "v01" |
| MCNN | 02 | color metadata, 16 batch size, 30 epochs, no augmentation |
| | 02.1 | Same as m02, but color metadata is scaled to [0, 1] |
| | 02.2 | Same as m02, but also with face metadata |
| | 02.3 | Same as m02.1, but also with face metadata |
| | 02.5 | Same as m02.1, but scaling face and text metadata as well |
| HCNN | 03 | 1 level, stratified, no metadata, 32 batch size, 30 epochs, no augmentation |
| XCNN | 04 | no metadata, all batch, 100 epochs, no aug, no es, used m01.1 |
| | 04.1 | Same as m04, but with sample weights |
| | 04.2 | Same as m04.1, but with Bayesian optimization and cross validation on train set |

Initial Learning Rate: 0.001
Adam Optimizer with
$\beta_1 = 0.9$, and $\beta_2$=0.999

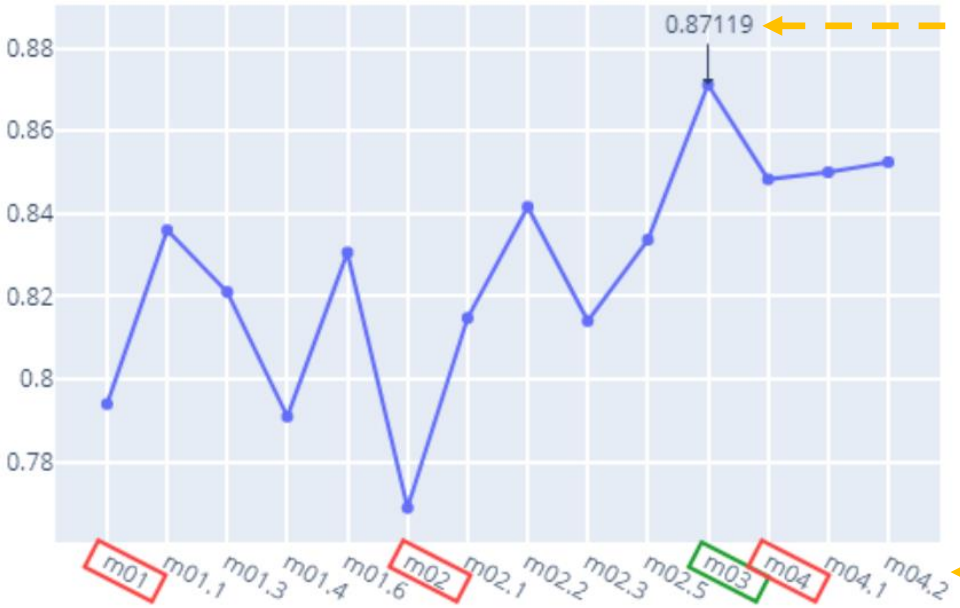
Learning Rate: 0.3

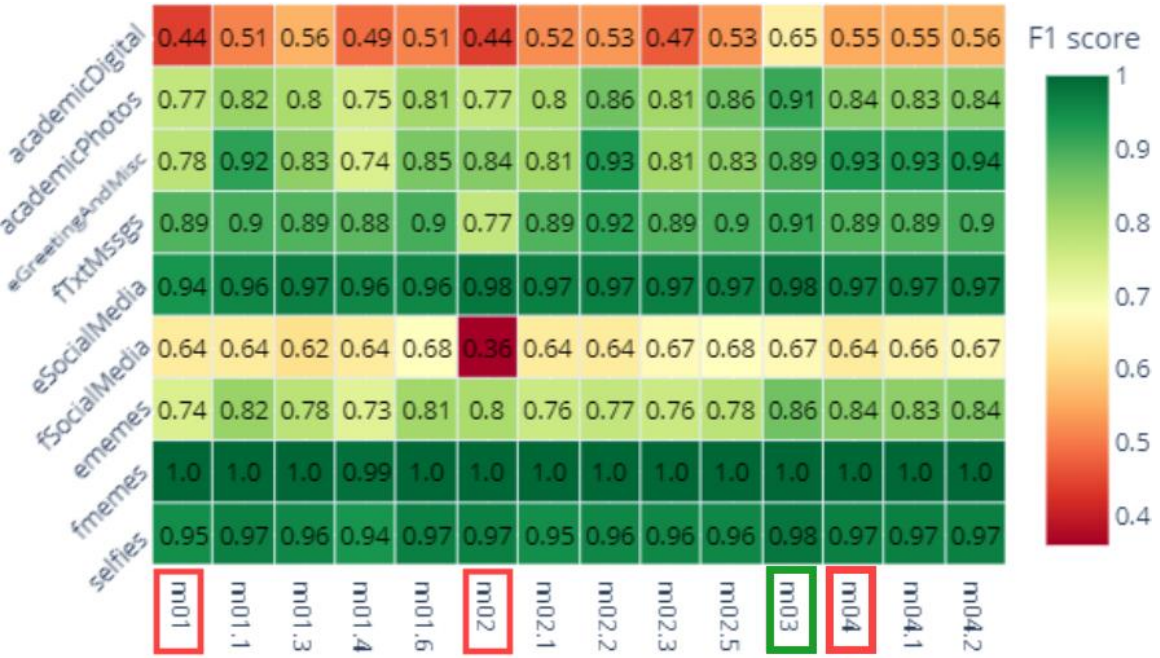
up next: Results

# Results & Analysis

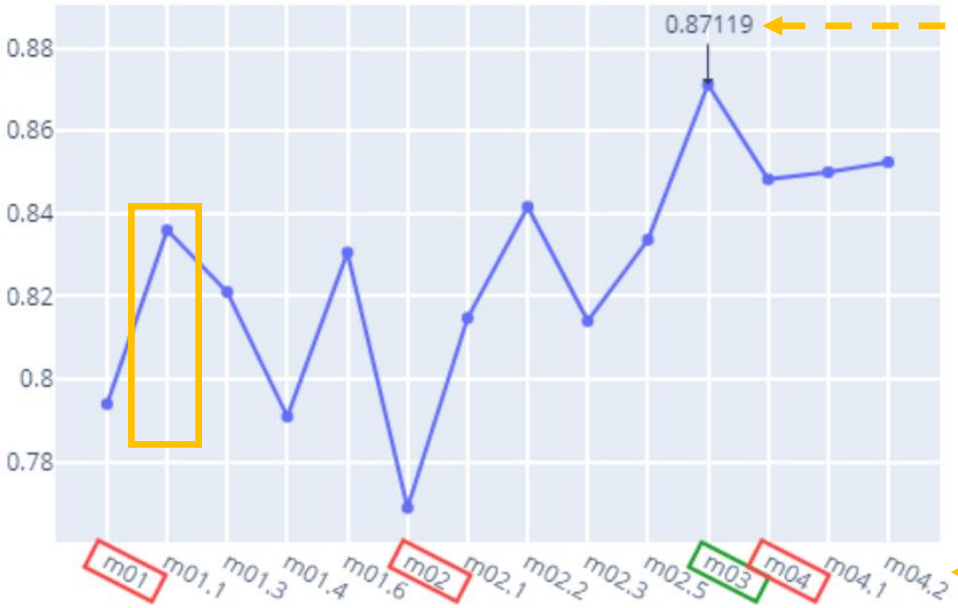
# Results

# Analysis & Recommendations



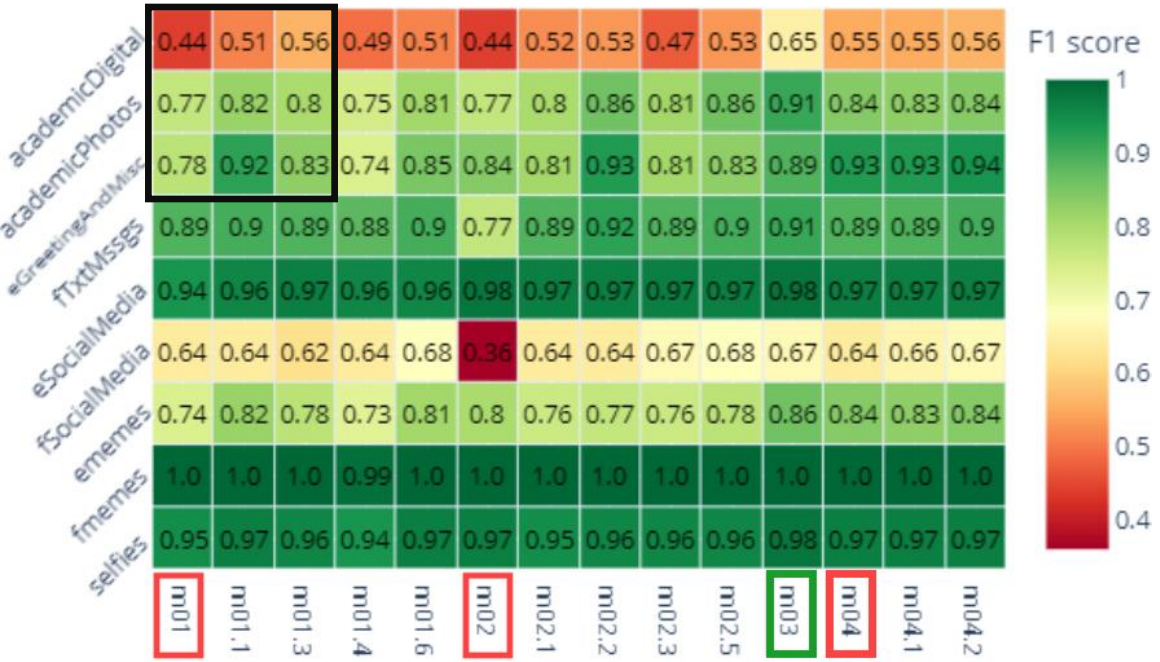



# Results




F1-Score Avg.

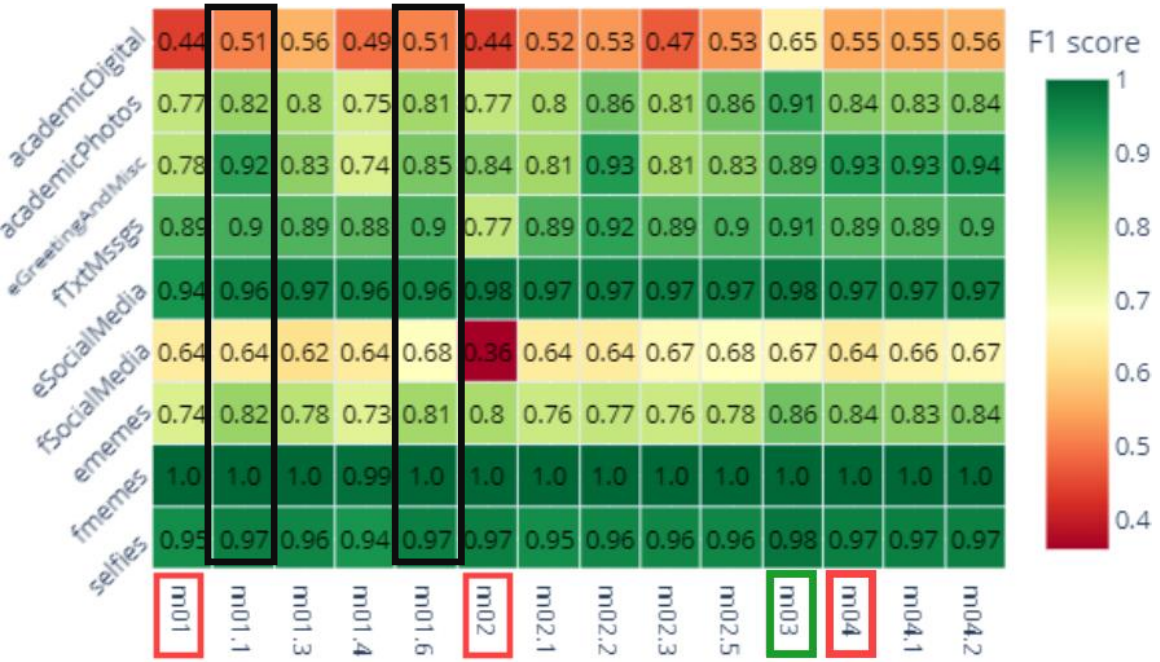
Model Version




# Analysis & Recommendations

1. batch size ↔ F1

# Results

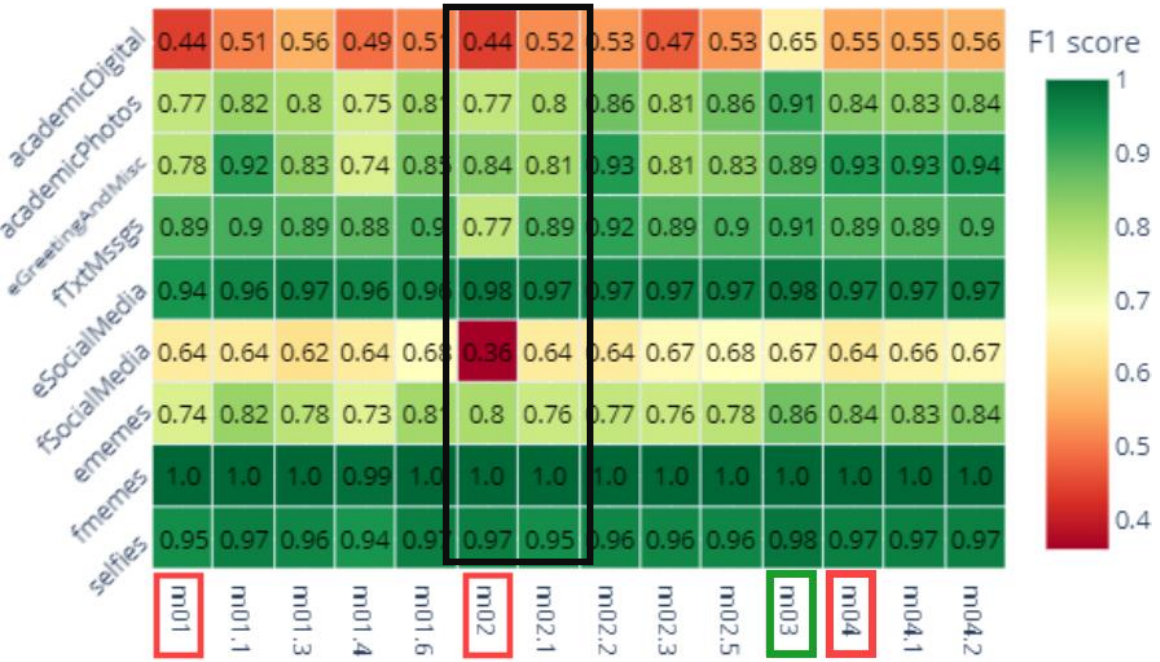




# Analysis & Recommendations

1. batch size ↔ F1

2. Removing outliers in Minorities ↔ F1

# Results

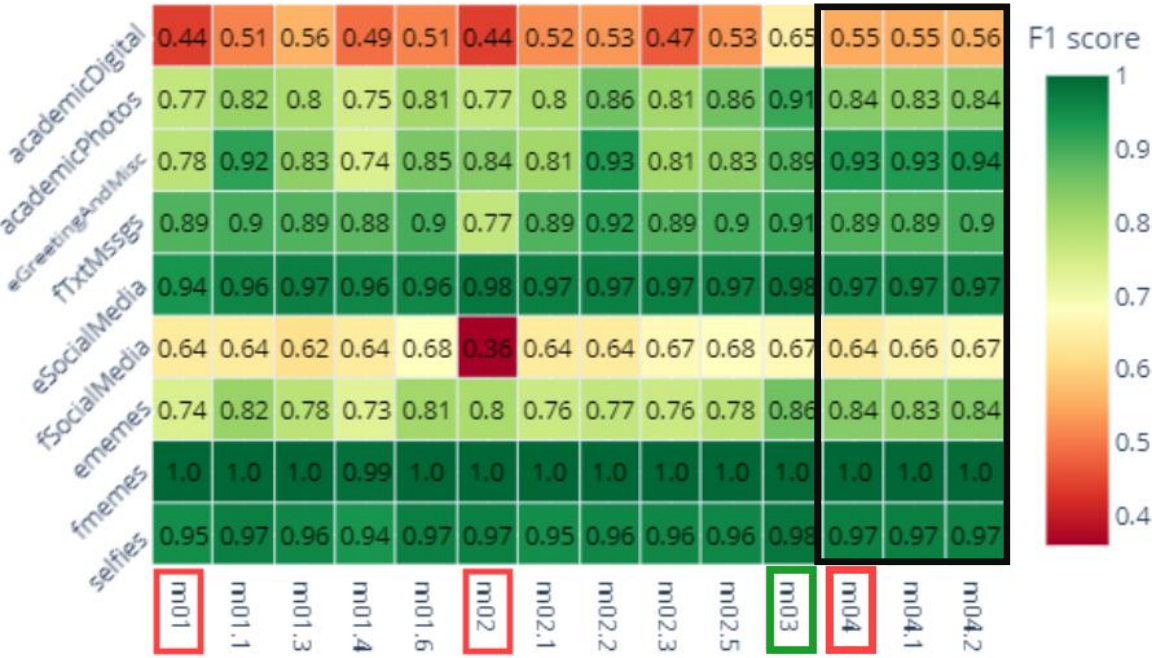

# Analysis & Recommendations

1. ↑ batch size ↔ ↑ F1

2. Removing outliers in Minorities ⇹ ↑ F1

3. Normalizing input, ex., colour metadata ↔ ↑ F1

# Results


# Analysis & Recommendations

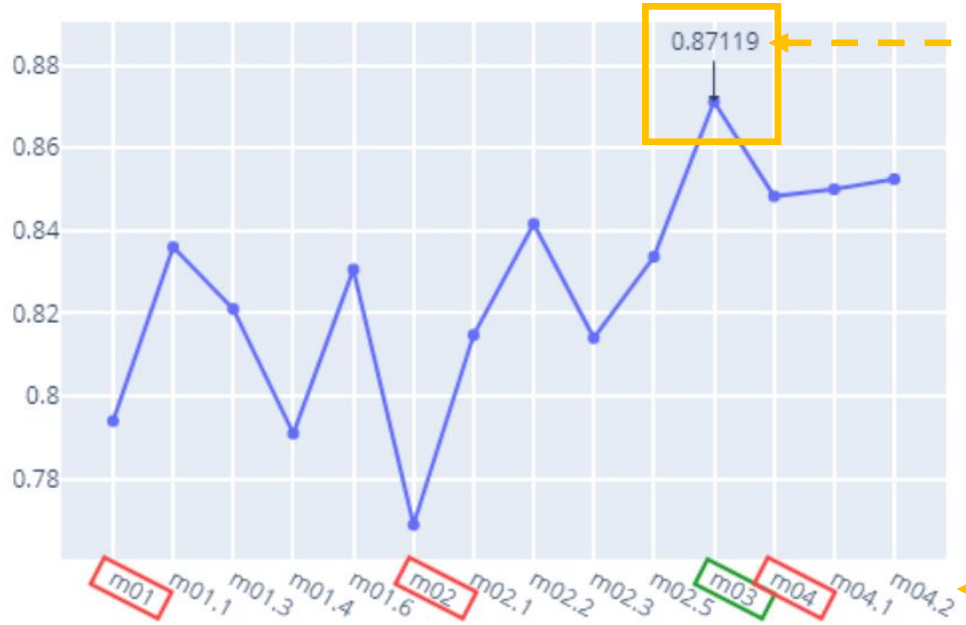
1. batch size ↔ F1

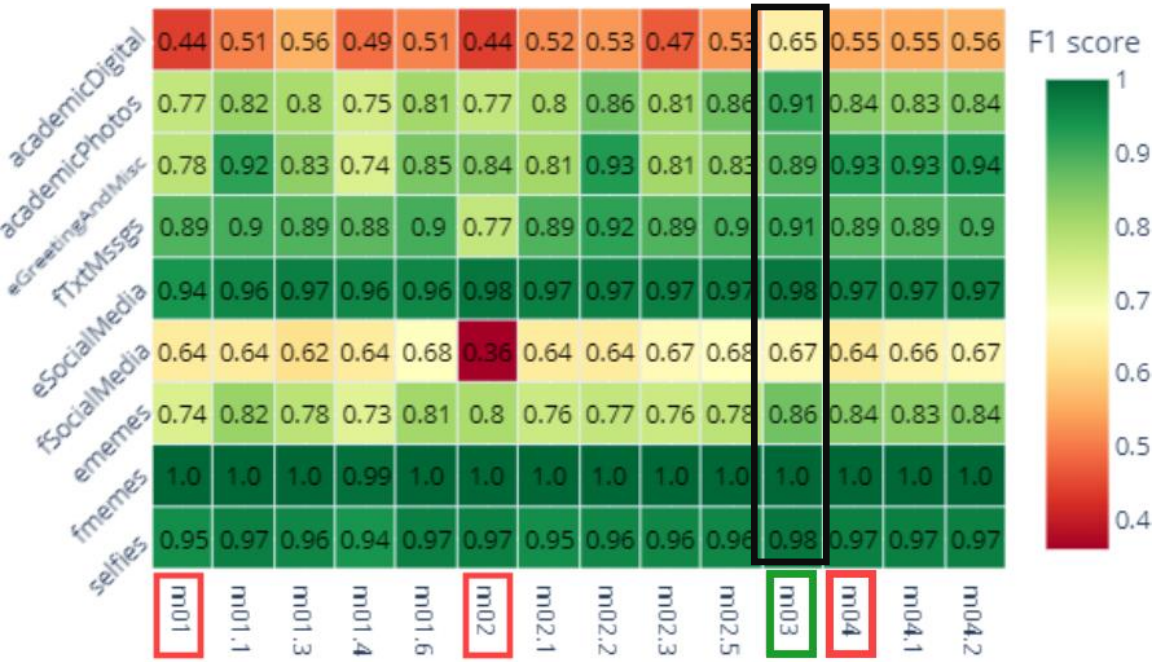
2. Removing outliers in Minorities ↔ F1

3. Normalizing input, ex., colour metadata ↔ F1

4. Sample weights, & Bayes. opt. ↔ F1 still overfitting

# Results




# Analysis & Recommendations

1. ↑ batch size ↔ ↑ F1

2. Removing outliers in Minorities ⇹ ↑ F1

3. Normalizing input, ex., colour metadata ↔ ↑ F1

4. Sample weights, & Bayes. opt. ↔ ↑ F1 still overfitting

5. Easy Classification (general) then harder classification (specific) ↔ ↑ F1

# Conclusion, Limitations, & Future Work

1. “Data-Centred” approach should be further applied due to using a custom dataset. Due to time constraints, it was difficult to keep manually cleaning images and performing extra data preparation steps

2. “Model-Centred” approach has also the disadvantage of time and computational restraints; Considerable amount of effort should go to profiling the model training process to minimize time and power consumption.

3. Outdated libraries, like RetinaFace & PaddleOCR, should be replaced to avoid dependency issues & ensure environment packages’ compatibility.

4. A basic mobile app should be developed to act as an easy interface for the user to categorize their private images.


End of Presentation
ANY QUESTIONS?
memecreator.org