Sharing Device identification on Images from Social Media Platforms

By:

Mohit (2022AIM1003)
Sazid Ali (2022AIM1013)
Ullas Shrivastava (2022CSM1015)

Problem Statement :

Given image from some social networking platform, predict the type of device through which it is being shared.

?

- ANDROID
- APP-MAC
- APP-WIN
- IPHONE
- ORIGINAL
- WEB-IPAD
- WEB-MAC
- WEB-WIN

MOTIVATION

Since when uploaded and shared via social networking platforms, images undergo strong processing such as JPEG compression and resizing, which substantially remove forensic traces. Indeed, to optimize transfer bandwidth as well as display quality, most Social Networks enforce their own compression/resizing policy, which is generally neither published nor fixed. Those concerns raise the need to identify the SN platform or sharing apps in order to recover partial knowledge.

DATASET : SHADE Dataset

- Taken 50 raw images and resized to three different resolutions (337 × 600, 1012 × 1800 and 1687×3000) with an aspect ratio of 9:16 and then JPEG-compressed using six quality factors (QF = 50, 60, 70, 80, 90, 100), yielding a total of 900 images.
- 8 classes (ANDROID, APP-MAC, APP-WIN, IPHONE, ORIGINAL, WEB-IPAD, WEB-MAC,WEB-WIN)

FEATURE EXTRACTION (DE-QUANTIZED DCT COEFFICIENTS)

We extract from each image the normalized histograms of de-quantized DCT coefficients, separately for each frequency, and we retain only the histogram bins corresponding to integer values in [−20, 20] and only for the first 9 AC frequencies in zigzag order. By concatenating the histograms, we obtain a $9 \times (2*20 + 1) = 369$ dimension signal-based feature vector.

FEATURE EXTRACTION (META-DATA)

We extract 182 features values from metadata of image.

1) Quantization tables : 128
2) Huffman Table values : 32
3) Component information for each YCbCr channel : 18
4) Optimized coding and progressive mode: 2
5) Image size: 2

Process to find De-Quantized DCT Coefficients

1. The process is done block by block as in JPEG compression
2. Convert the Image in Grayscale if available in colour
3. Subtract 128 from each pixel value
4. Find DCT of the block
5. Divide the resultant with the quantization matrix and round the results
6. Now again multiply the resultant with the quantization matrix
7. Resultant block contains the de-quantized DCT coefficients

Feature Extraction

DC	AC1	AC5	AC6				
AC2	AC4	AC7					
AC3	AC8						
AC9							

1. These 9 AC components of de-quantized DCT are extracted for each block of the image
2. Took AC1 of all the blocks and created a histogram in [-20, 20] range consisting of 41 bins.
3. Similarly the histogram is created for all 9 AC frequencies
4. Finally all the histograms are concatenated to create a feature vector of 369 dimensions.

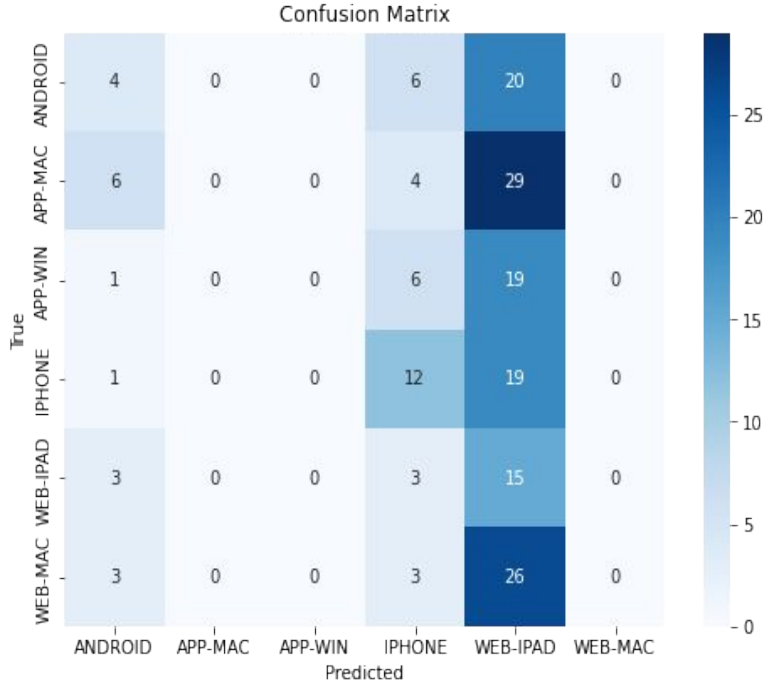
Models Tried

1. SVM
2. Multi Level Perceptron
3. P-CNN
4. Random Forest

SVM

1. Used data with quality factor = 50
2. With 9 AC Components
3. Meta Data not included
4. Two labels skipped: Original and WEB-WIN
5. Testing accuracy - 17.7%

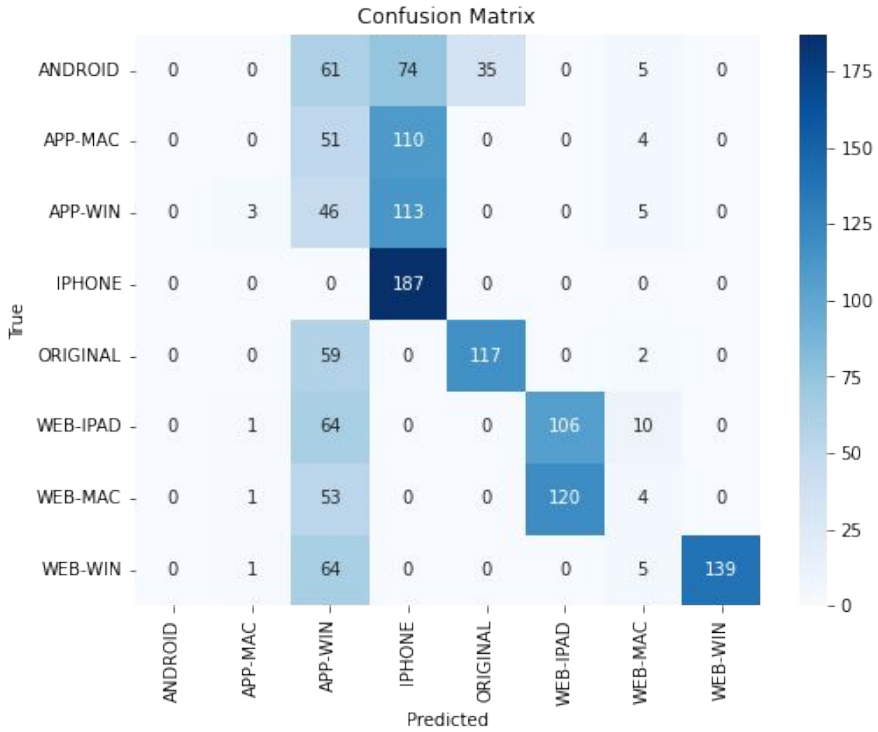
Observations: The model confuses a lot and does not perform acceptably



SVM

1. Used images from all QF-* folders
2. With 9AC components
3. Testing Accuracy = 41.59%

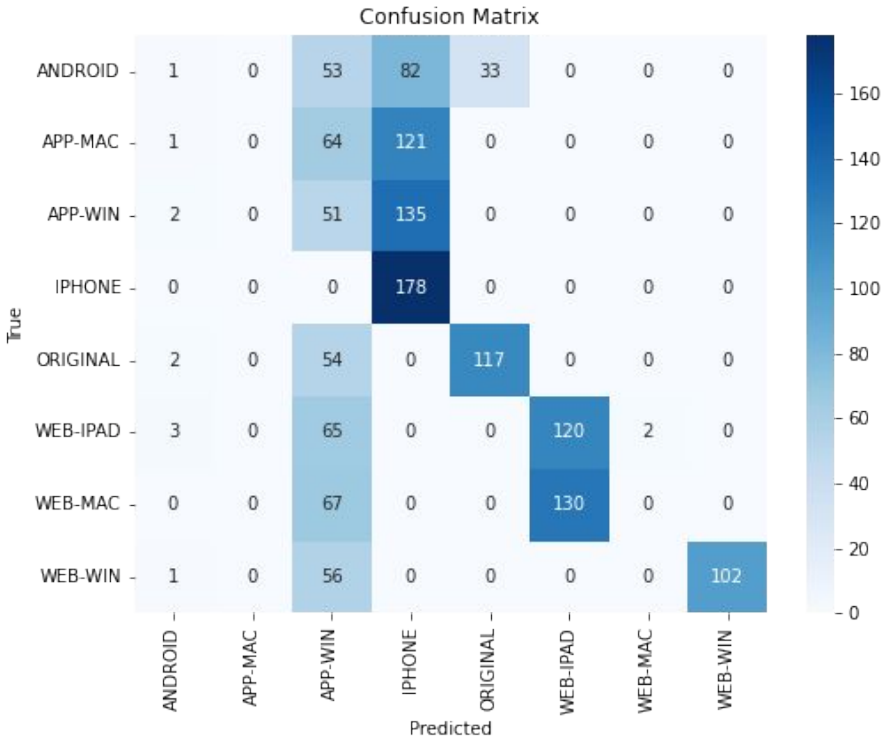
Observations: Model performs better compared to when the data was less and gives a good accuracy for iPhone. But overall performance is still poor



SVM

1. Used images from all QF-* folders
2. Used all AC Components (Feature vector)
3. Testing Accuracy = 39.51%

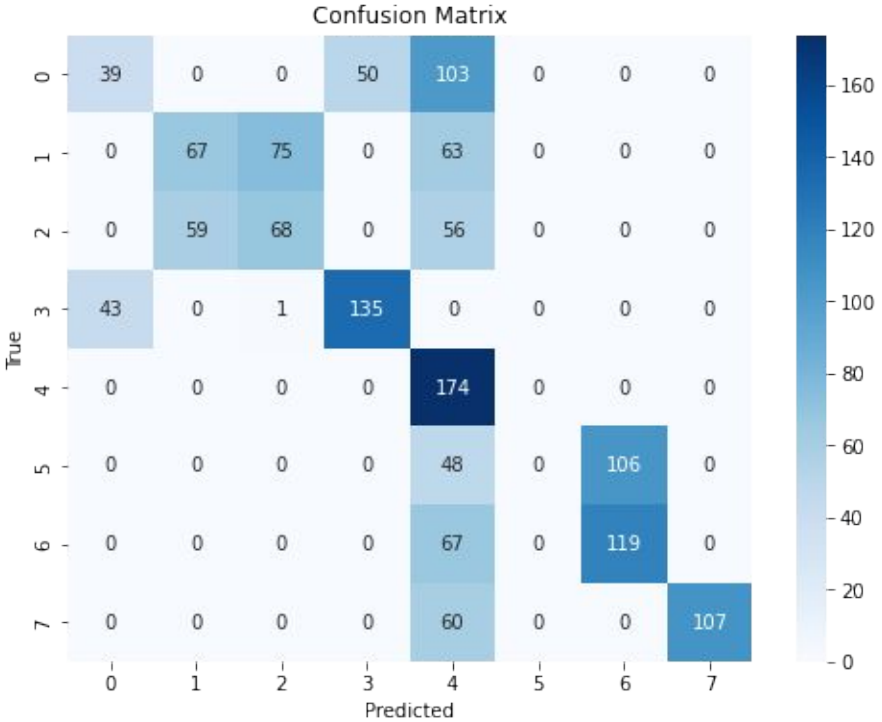
Observations: Accuracy dropped when all the components are included in feature set. But iPhone's accuracy is still good.



P-CNN

1. Used images from all QF-* folders
2. Used 9 AC Components (Feature vector)
3. Model - CNN (LR=0.0001)
4. Testing Accuracy = 49.24%

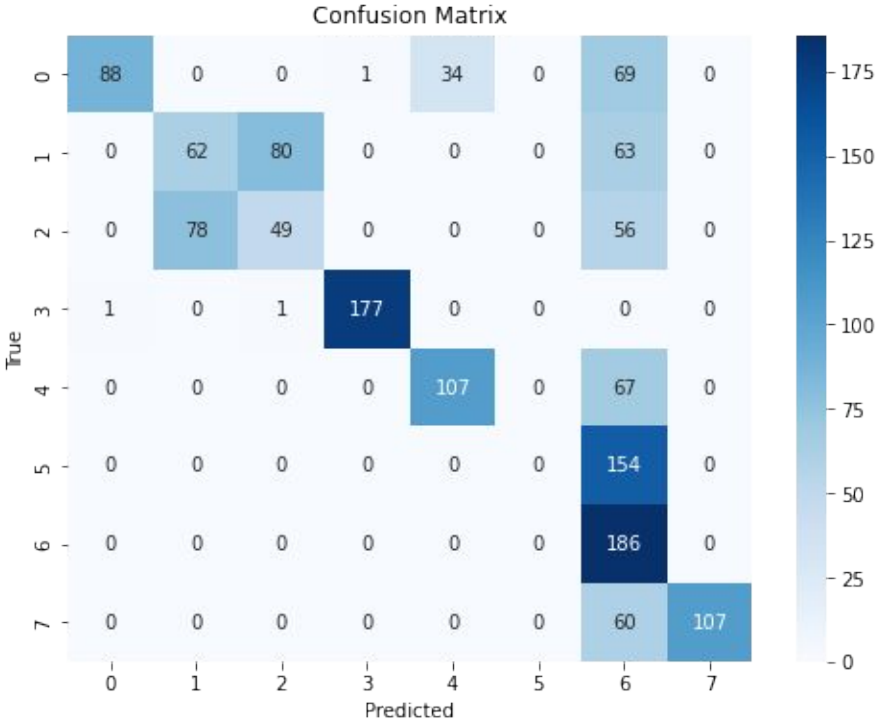
Observations: Accuracy of iPhone & Original improved. There is slight improvement in the categorical accuracy but overall model performs bad.



P-CNN

1. Used images from all QF-* folders
2. Used all AC Components (Feature vector)
3. Model - CNN (LR=0.0001)
4. Testing Accuracy = 53.89%

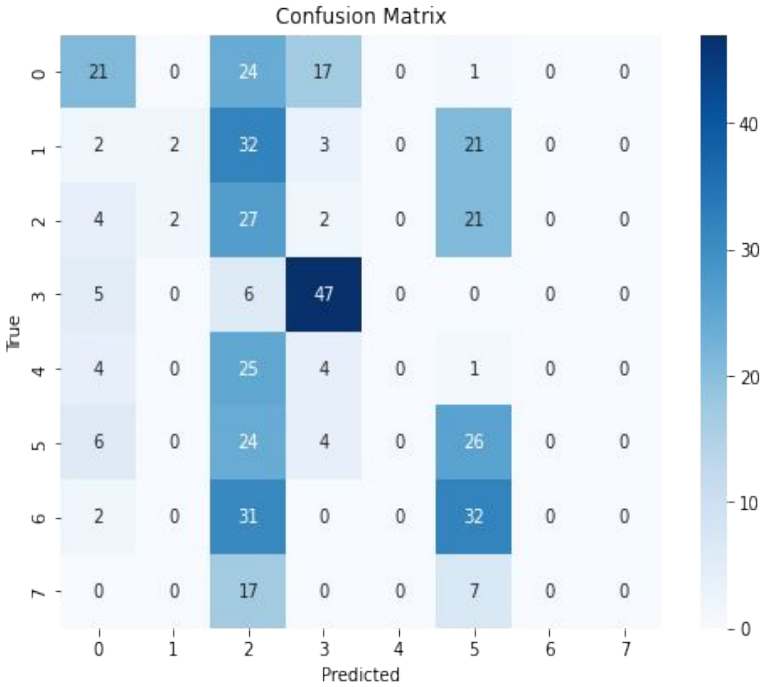
Observations: Increasing the number of components increased accuracy little bit. But not much significant change. Accuracy for iPAD is increased. Overall, model performs well compared to other model.



Multi Level Perceptron

1. Used images from QF-50 and QF-100
2. With 9 AC components
3. Meta data not included
4. Testing Accuracy = 29.2%
5. Architecture:
    a. Layer 1: 256
    b. Layer 2: 128
    c. Layer 3: 64
    d. Layer 4: 64
    e. Layer 5: 32
    f. Layer 6: 32
6. Activation is ReLu in all

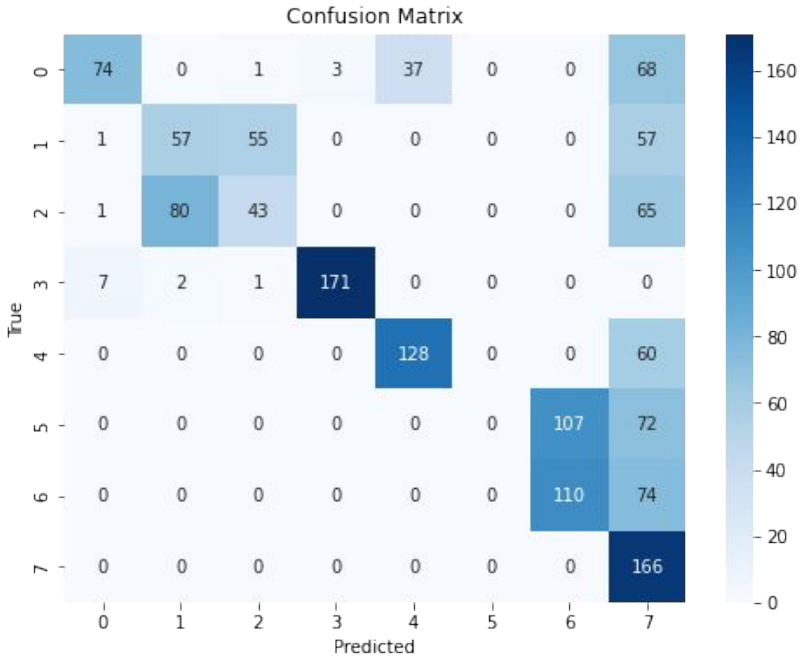
Observations: Model performs badly but gives a good accuracy for iPhone



Multi Level Perceptron

1. Used images from all QF
2. With 9 AC components
3. Testing Accuracy = 52.01%
4. Architecture:
   a. Layer 1: 256
   b. Layer 2: 128
   c. Layer 3: 64
   d. Layer 4: 64
   e. Layer 5: 32
   f. Layer 6: 32
5. Activation is ReLu in all

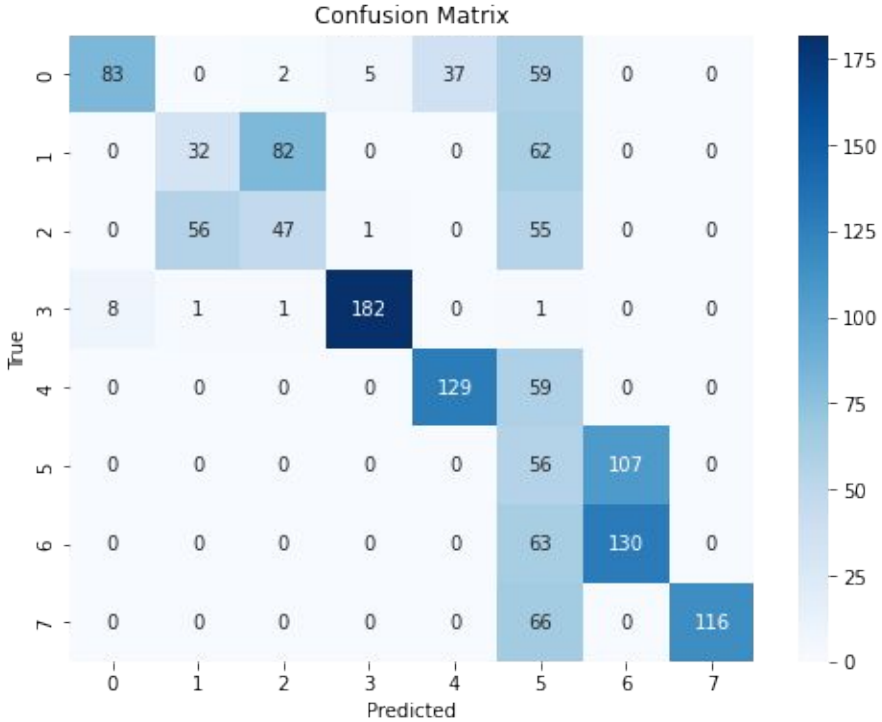
Observations: Model performs better when all data is considered and gives a good accuracy for iPhone, Original and WEB-WIN. But overall performance is still poor



Multi Level Perceptron

1. Used images from all QF-* folders
2. With all AC components
3. Testing Accuracy = 53.8%
4. Architecture:
    a. Layer 1: 256
    b. Layer 2: 128
    c. Layer 3: 64
    d. Layer 4: 64
    e. Layer 5: 32
    f. Layer 6: 32
5. Activation is ReLu in all

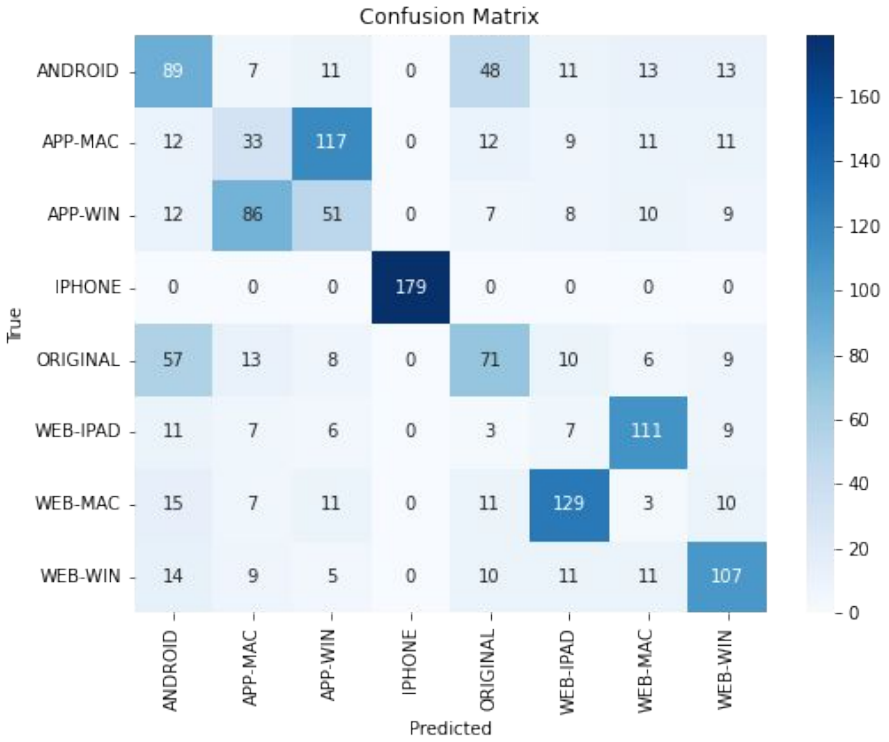
Observations: No Significant improvement over the previous one



Random Forest

1. Used images from all QF-* folders
2. With 9 AC components
3. Testing Accuracy = 37.5%

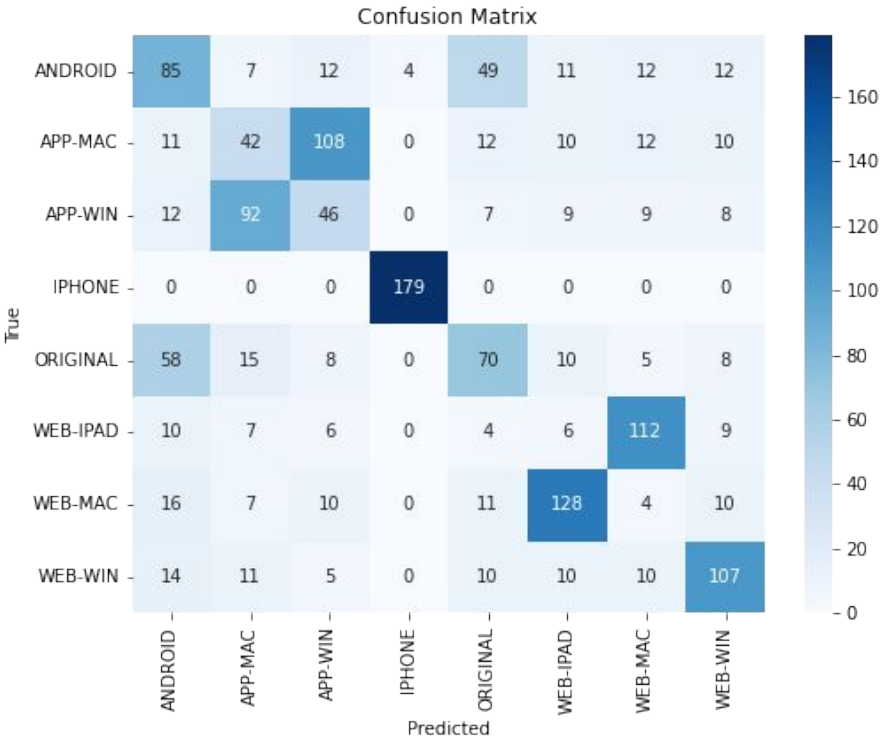
Observations: Accuracy for iPhone is 100% and Accuracy for WEB-WIN is also very good. But overall performance is poor



Random Forest

1. Used images from all QF-* folders
2. With all AC components
3. Testing Accuracy = 37.43%

Observations: No Significant change in performance even after including all AC components



Comparison of all Models

Model	With 9 AC Components	With all AC Components
SVM	41.59	39.51
MLP	52.01	**53.81**
P-CNN	49.24	**53.89**
Random Forest	37.50	37.43

CONCLUSION

We got best accuracy as 53.8% using Multi Layered Perceptron and 53.89% using P-CNN. In this model we used 9 AC component of de-quantized DCT coefficients of images and metadata of images. Also, we used images from all QF-* folders of all classes of image.

In all the classes, we got best accuracy for iPhone and iPAD as maximum as 100%.

In the research paper, best accuracy mentioned is 72%.

CHALLENGES

1. Feature extraction is a time consuming process as it involves calculation of DCT of the image.
2. Extracting metadata of image was challenging task because all social media platforms scraps the metadata (EXIF file) before the image is uploaded to the platform.
3. Dataset size was small. Only 50 photos was taken and it was used with different resolution to increase dataset size.

References

1. A. Tomasoni, S. Verde and G. Boato, "Sharing Device Identification on Images from Social Media Platforms," *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, Shanghai, China, 2022, pp. 1-6, doi: 10.1109/MMSP55362.2022.9948824.
2. Q. -T. Phan, G. Boato, R. Caldelli and I. Amerini, "Tracking Multiple Image Sharing on Social Networks," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 8266-8270, doi: 10.1109/ICASSP.2019.8683144.
3. Q. -T. Phan, C. Pasquini, G. Boato and F. G. B. De Natale, "Identifying Image Provenance: An Analysis of Mobile Instant Messaging Apps," *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, Vancouver, BC, Canada, 2018, pp. 1-6, doi: 10.1109/MMSP.2018.8547050.

THANK YOU