# Image Classification with JPG Data

**CSE 496**

**3rd Meeting**

**Esra Eryılmaz**

**Project Advisor: Prof. Dr. Yusuf Sinan Akgül**

**January 2023**

# Content

- Project Description
- Project Design Plan
- Environment
- Dataset
- Image compression method : DCT
- Project Architecture : CNN Model
- Training Results
- Success Criteria
- User Interface
- Resources

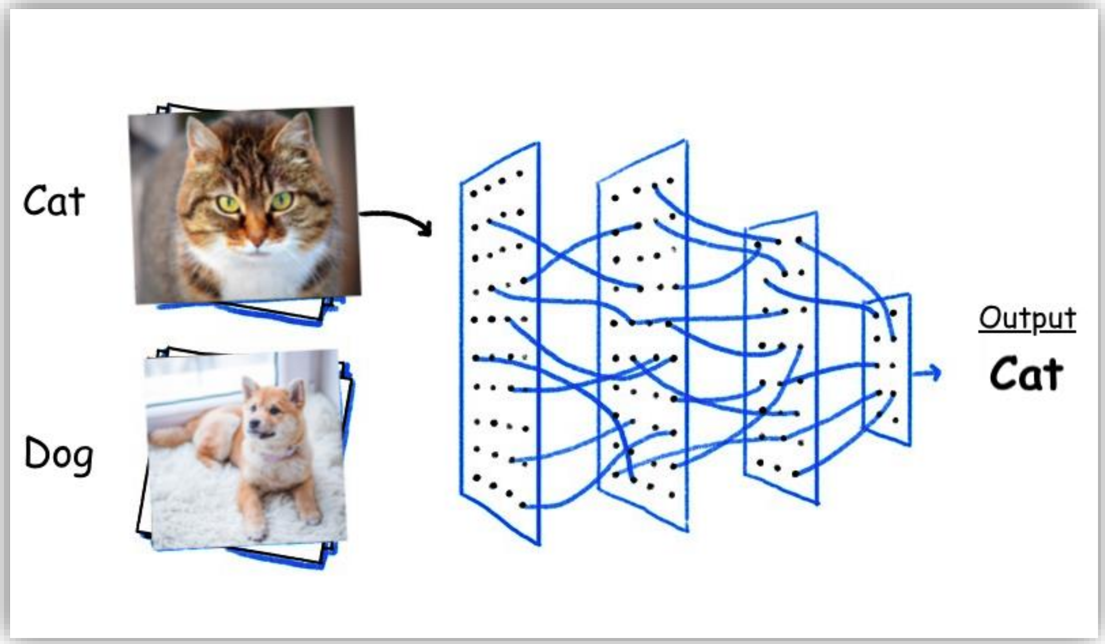
# Project Description

**Description :** Image Classification project with compressed images.

- In some applications images are compressed either for storage savings or fast transmission.
- Therefore a time consuming image decompression step is compulsory in order to apply the deep learning models.

## Project goal :

To eliminate this drawback, we will modify the well-known deep learning models to recognize the objects in compressed images.



# Project Design Plan

1st Meeting 26 Oct 2022

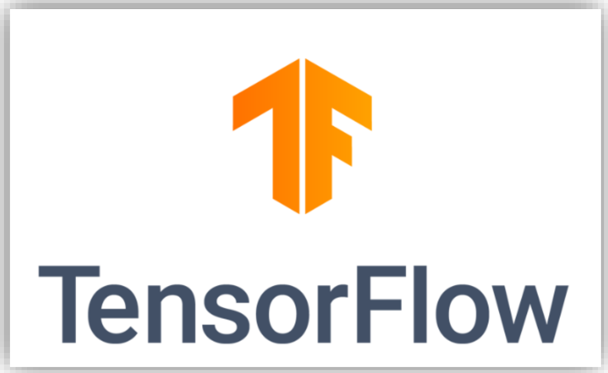
2nd Meeting 7 Dec 2022

TODAY 3rd Meeting 18 Jan 2023

Demo 19 Jan 2023

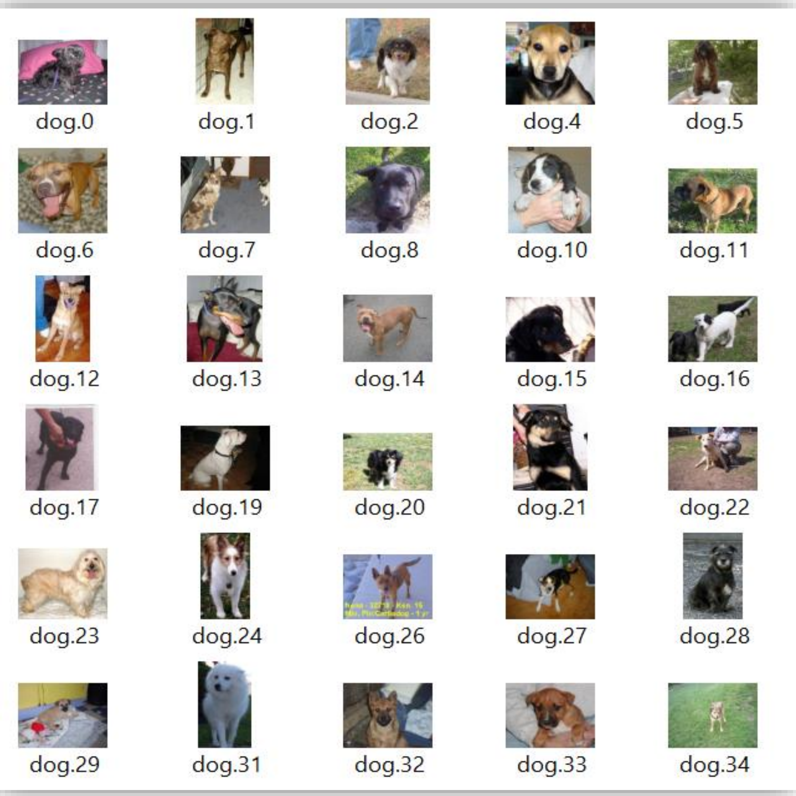
| Months | October 2022 | | | November 2022 | | | | December 2022 | | | | January 2023 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***Weeks*** | 17 Oct | 24 Oct | 31 Oct | 7 Nov | 14 Nov | 21 Nov | 28 Nov | 5 Dec | 12 Dec | 19 Dec | 26 Dec | 2 Jan | 9 Jan | 16 Jan | 23 Jan |
| *Literature review (image classification, image compression, deep learning)* | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | | | | |
| *Preparing the development environment,<br>learning the necessary libraries* | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | | |
| *Starting to apply deep learning models* | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | |
| *Comparing and analysing applied deep learning models* | | | | | | | | | ■ | ■ | ■ | ■ | | | |
| *Creating a simple UI for the demo* | | | | | | | | | | | | ■ | ■ | | |
| *Making the finishing touches, preparing the report and trailer* | | | | | | | | | | | | ■ | ■ | | |

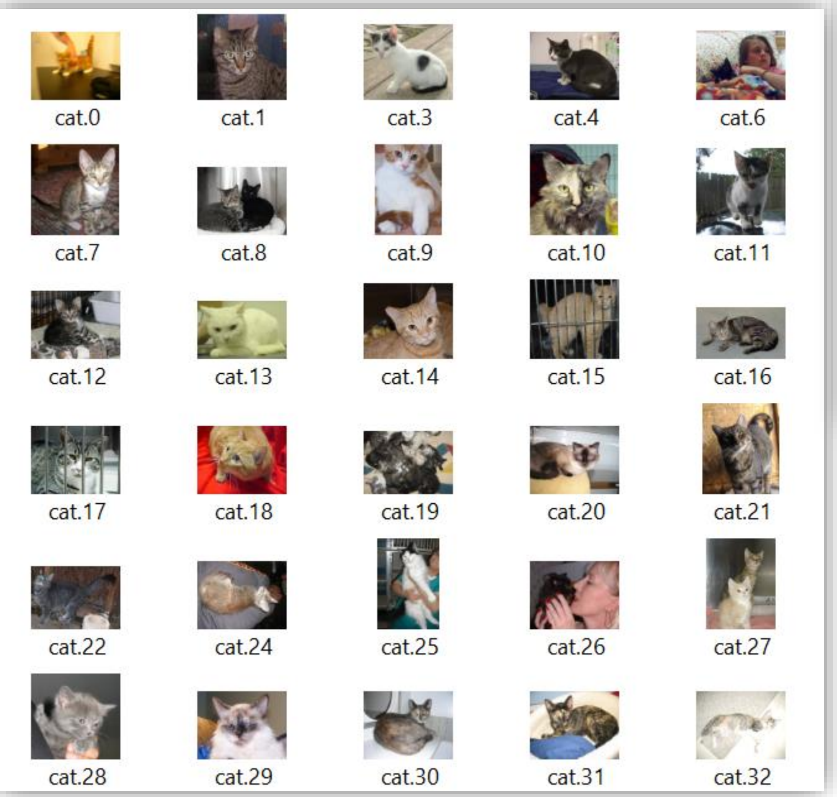
# Environment



# Dataset

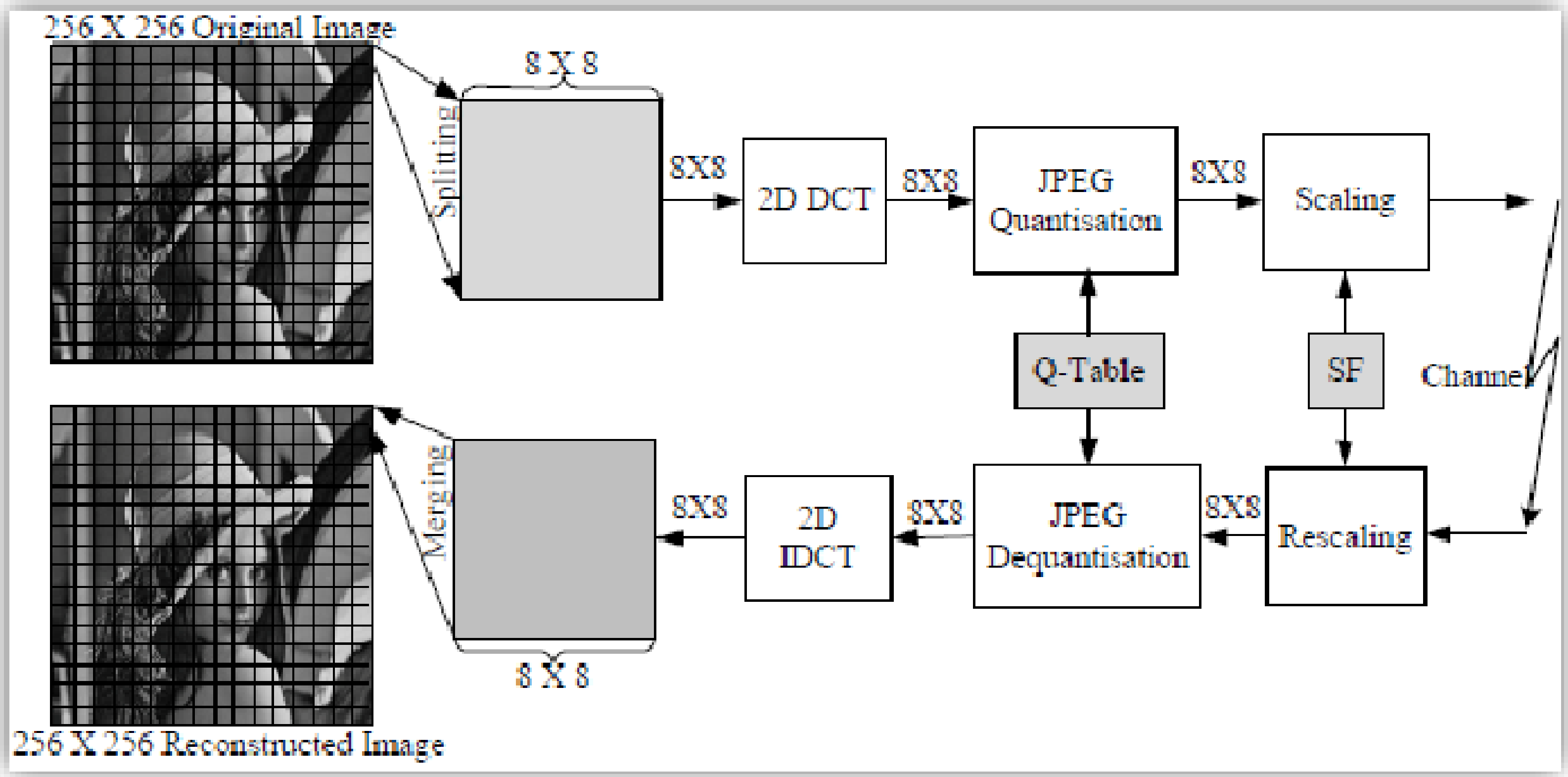
- **30.000** images
- 2 class





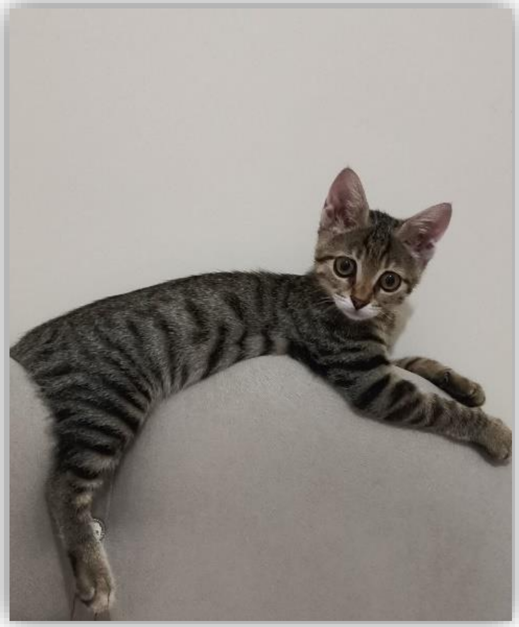
# Image Compression Method : **DCT**

- A **discrete cosine transform** is a math process that can be used to make things like MP3s and JPEGs smaller.
- It does this by breaking the sound or picture into different frequencies.



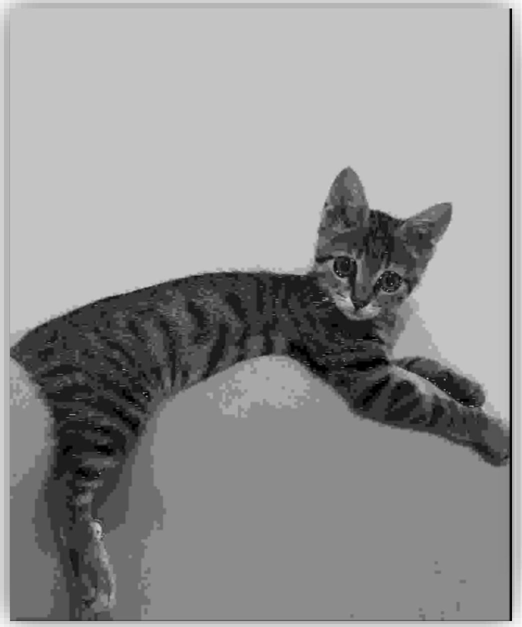
# DCT Output Example

## Original image :

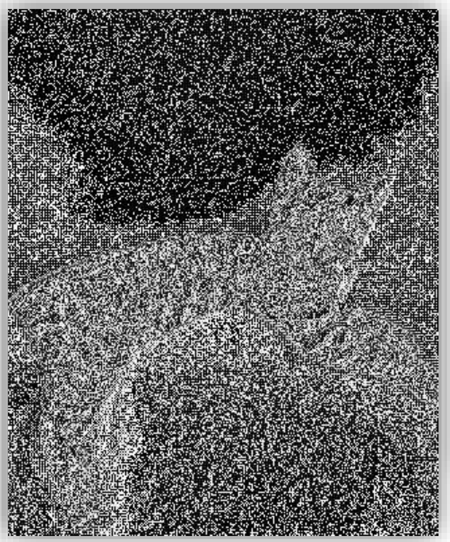


## DCT Outputs :

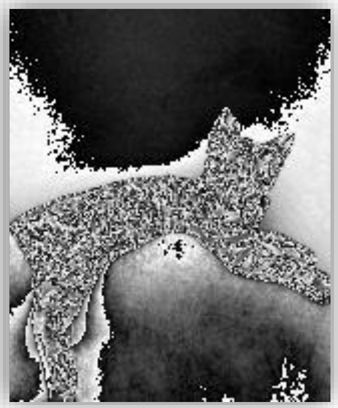
$$\frac{\text{dimension}}{1}$$



$$\frac{\text{dimension}}{4}$$



$$\frac{\text{dimension}}{8}$$



# Project Architecture: **CNN Model**

- A **convolutional neural network** (CNN) is a network architecture for deep learning that learns directly from data.

- CNNs are particularly useful for finding patterns in images to recognize objects, classes and categories.

- The output from our CNN model is either 1 or 0

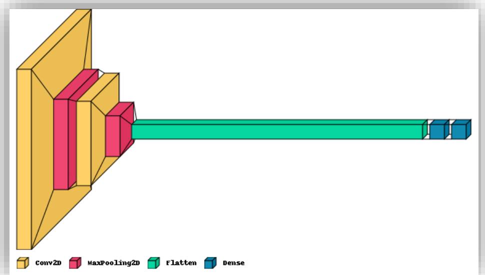
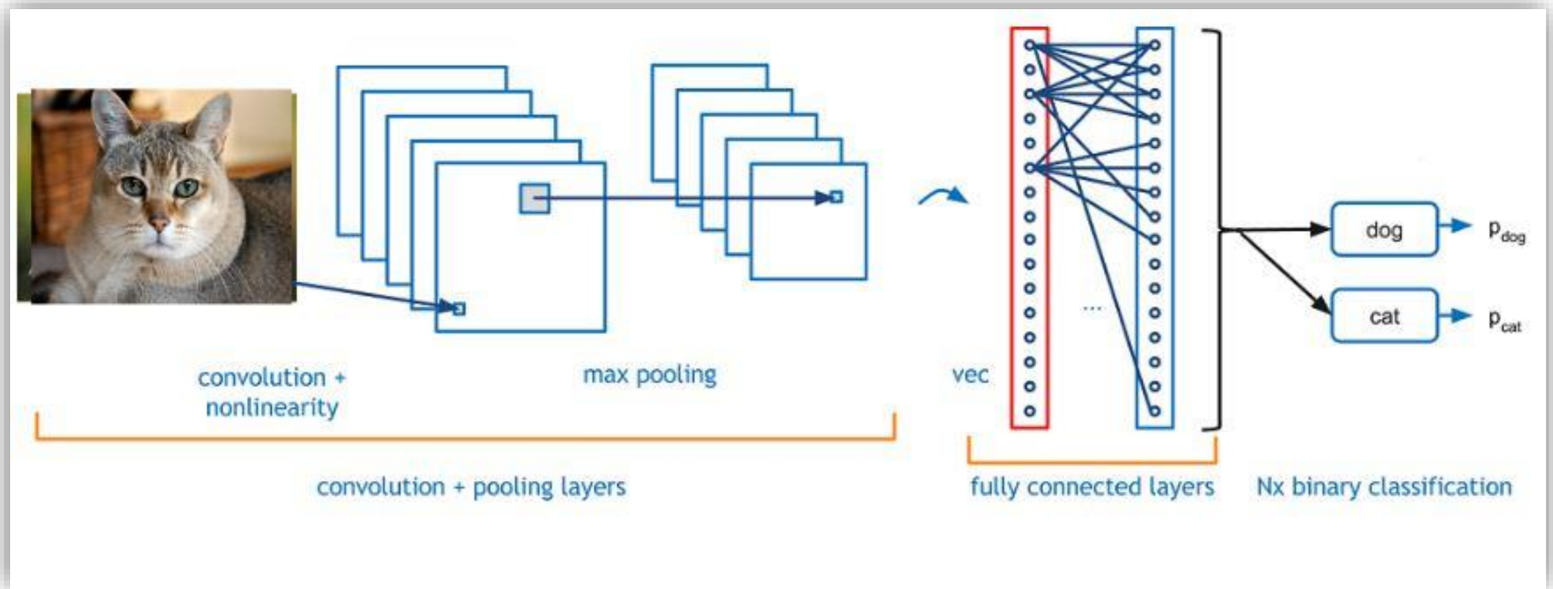


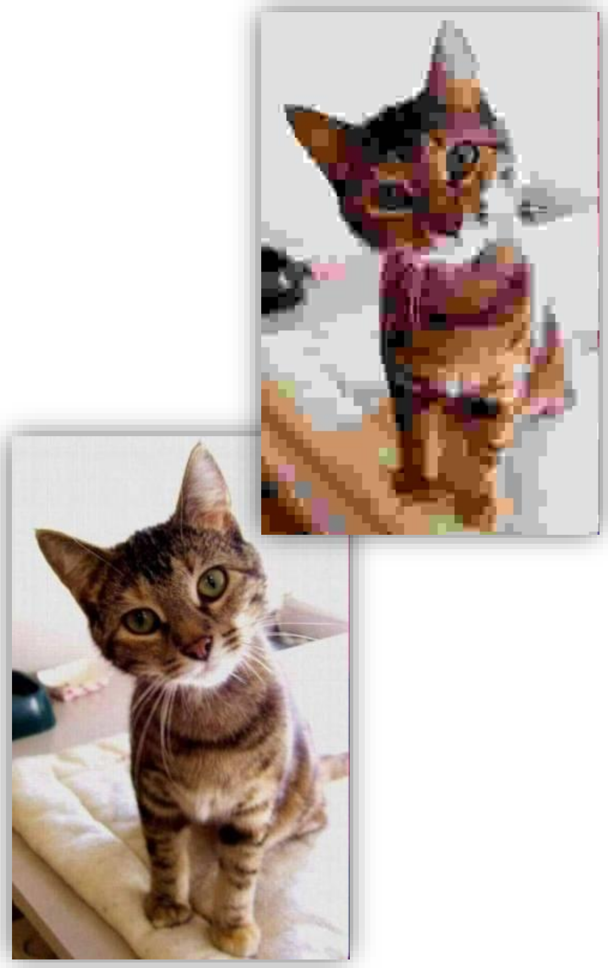
**Figure :** Layer visualization of my model



# Training Results

- Results that I trained by simply reducing the **quality** :
  (before image compression with DCT)

| Image Quality | Accuracy |
|---|---|
| **%100** | 0.893 |
| **%50** | 0.890 |
| **%30** | 0.888 |
| **%10** | 0.880 |
| **%1** | 0.902 |



# Training Results

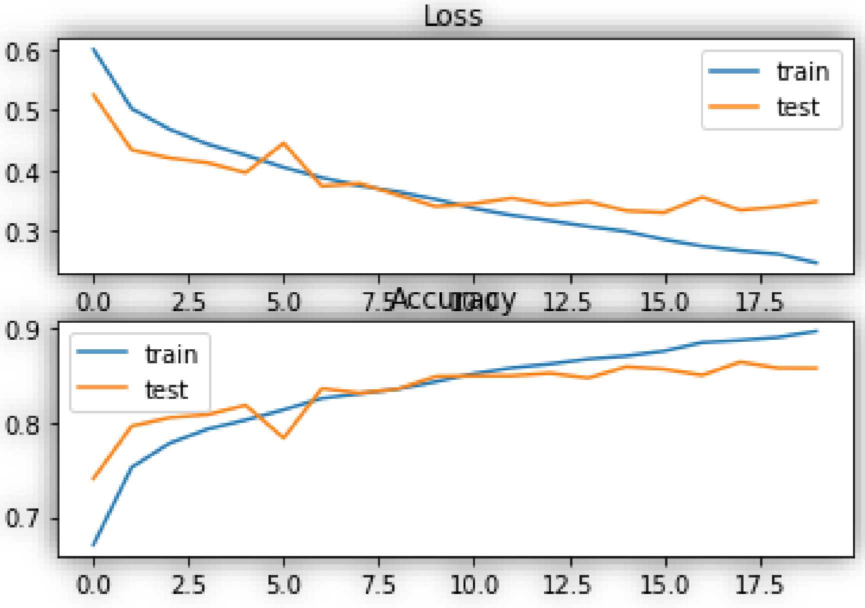
Comparison to the original :

- Accuracy : **-15%**
- Time : **1.8x faster**

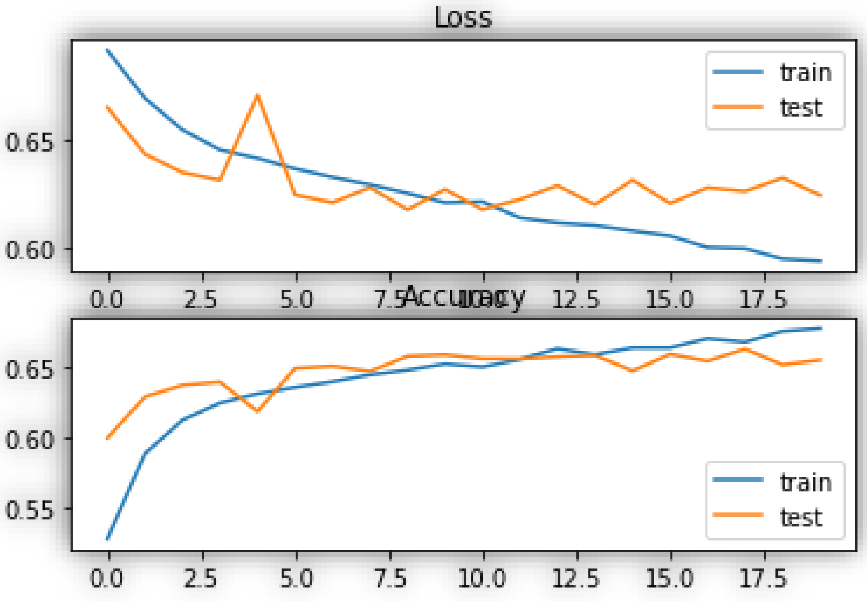
| | **Accuracy** | **Training time** |
|---|---|---|
| **Original images** | 30 epoch → 0.92<br>→ 0.91 | 159 min<br>200 min |
| | 20 epoch → 0.89 | 167 min |
| **Images with DCT (dimension/8)** | 30 epoch → 0.77<br>→ 0.68<br>→ 0.76 | 88 min<br>99 min<br>108 min |
| | 25 epoch → 0.70 | 24 min |
| | 20 epoch → 0.67 | 19 min |
| **Images with DCT (dimension/4)** | 30 epoch → 0.71<br>→ 0.70 | 115 min<br>167 min |
| | 20 epoch → 0.68 | 153 min |

# Training Results

- With original images



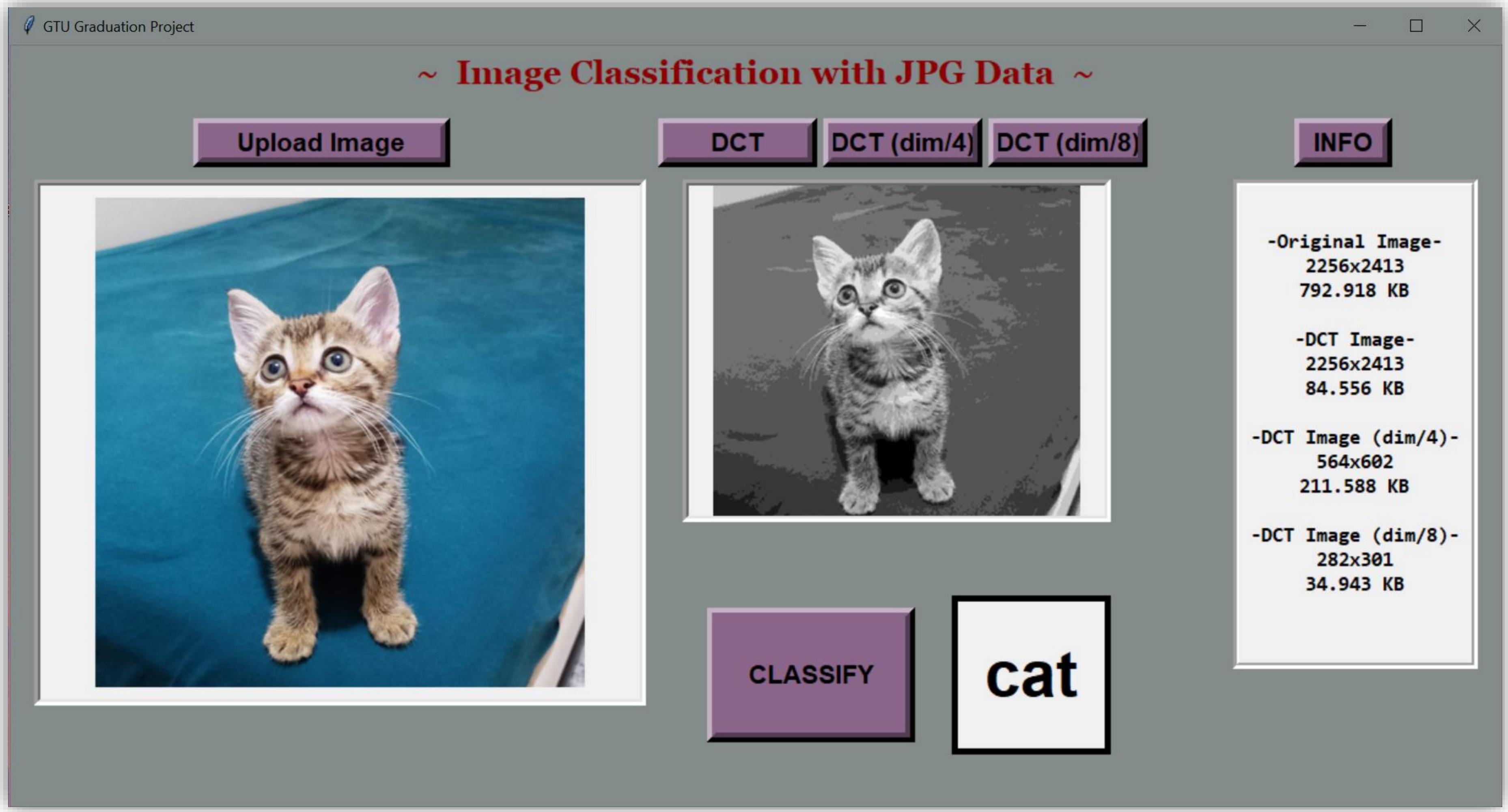
- With DCT applied images



# Success Criteria

- ✗ The average accuracy difference between RGB and JPG models should be no more than -5% loss. (< -5%)
- ✓ The model with JPG is about 1.5× faster than regular RGB images. (> 1.5x)
- ✓ At least 30.000 data will be used. (= 30.000)

# User Interface



-Original Image-
2256x2413
792.918 KB

-DCT Image-
2256x2413
84.556 KB

-DCT Image (dim/4)-
564x602
211.588 KB

-DCT Image (dim/8)-
282x301
34.943 KB

# Resources

[1] Seales, W. Brent, et al. "Object recognition in compressed imagery." Image and Vision Computing 16.5 (1998): 337-352, doi.org/10.1016/S0262-8856(97)00072-3.

[2] Gueguen, Lionel, et al. "Faster neural networks straight from jpeg." Advances in Neural Information Processing Systems 31 (2018).

[3] B. Deguerre, C. Chatelain and G. Gasso, "Fast object detection in compressed JPEG Images," 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 333-338, doi: 10.1109/ITSC.2019.8916937.

[4] Das, Nilaksh, et al. "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression." arXiv preprint arXiv:1705.02900 (2017).

[5] Evans, R. David, Lufei Liu, and Tor M. Aamodt. "Jpeg-act: accelerating deep learning via transform-based lossy compression." 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2020.