Class Activity: Image Preprocessing and CNN

Course: CS460 AI

Week 9: Computer Vision and Convolutional Neural Networks (CNNs)

Name: _Jetsada Wijit_

Student ID: _1640705339_

Section: _327G_

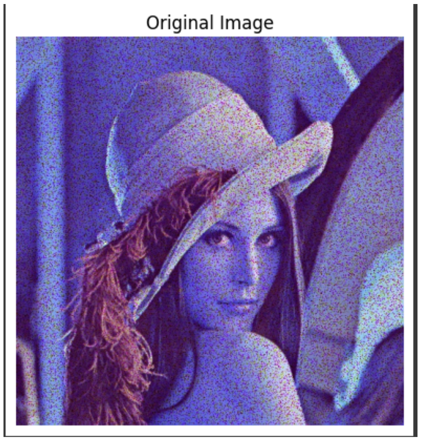
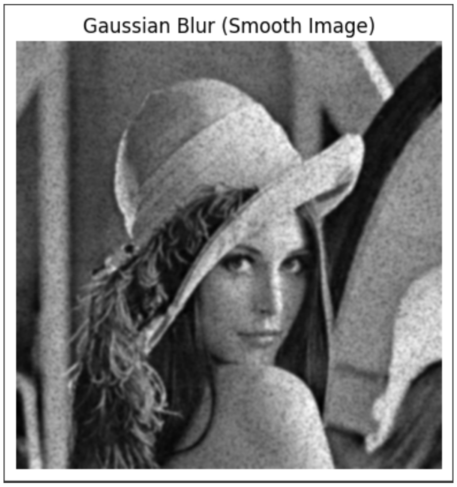
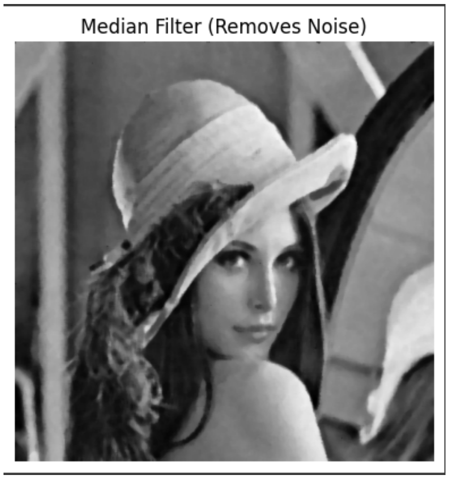
Activity: Capture Key Outputs & Answer Questions

Google Colab Notebook Link:
[https://colab.research.google.com/drive/1jJolaLxAeKjxSinS-1uqlncAShbPgNZK?usp=sharing]
Data Link: [Lena.jpg]

Part 1: Experiment with Edge Detection & Noise Filtering

Task: Compare “**Lena”** Original image with Gaussian Blur and Median Filter.

1. Which one is better for removing noise in Lena pictures? _Median_

Original	Gaussian	Median
		

Part 2: CNN

Step 1: Load and Inspect the Dataset

Write your observations:

1. What dataset do we use to train CNN models?

   cifar10

2. How many **classes (categories)** are in the dataset?

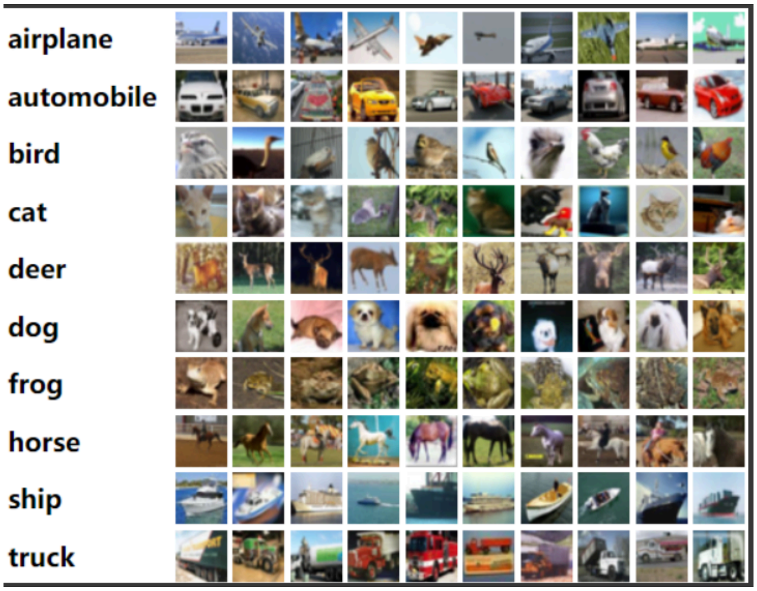
   10 class

3. What is the size of the train and test dataset?

   Train dataset size: 50,000

   Test dataset size: 10,000

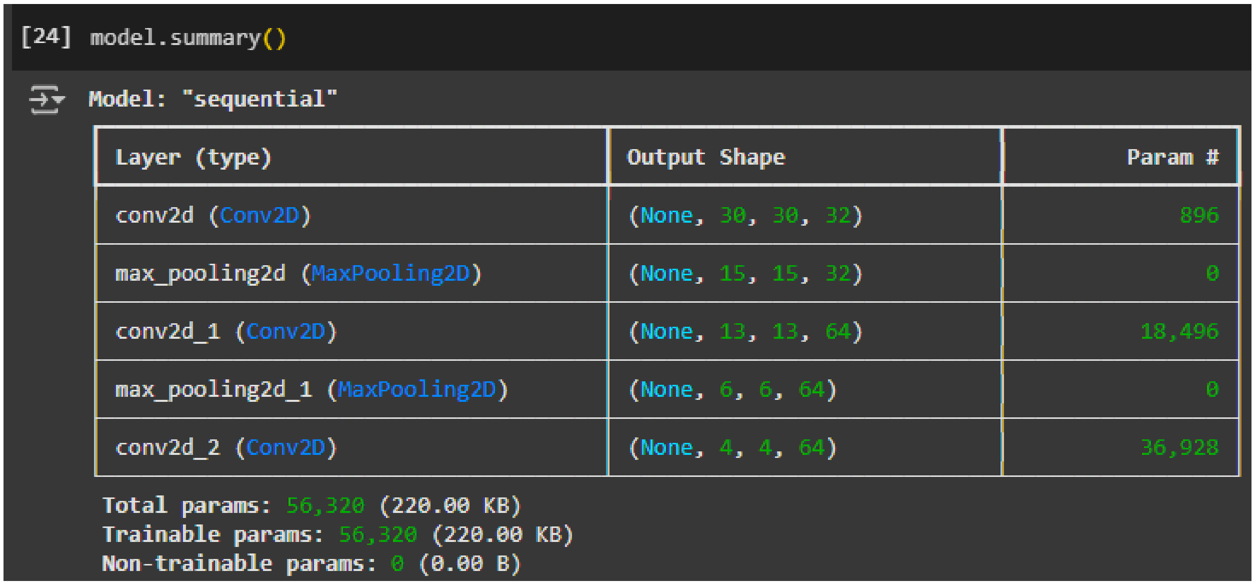
4. Capture example image of dataset



Step 2: Defining the CNN Architecture

Task: Build a CNN model using **Conv2D, MaxPooling, Flatten, Dense layers**.

1. Capture final model.summary() table



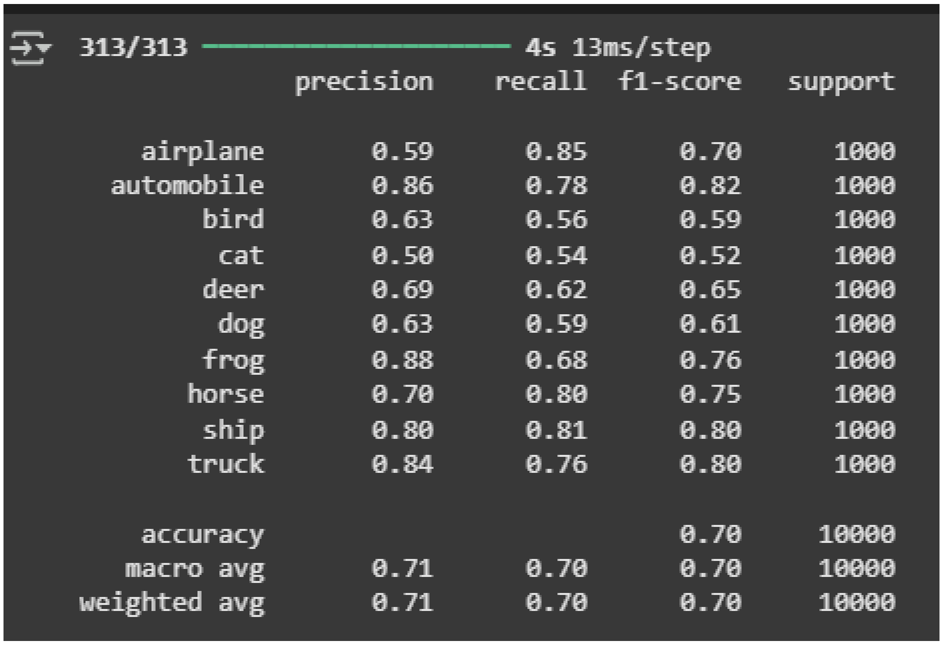
```
[24] model.summary()

Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━┓
┃ Layer (type)                      ┃ Output Shape            ┃        Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                   │ (None, 30, 30, 32)      │            896 │
├───────────────────────────────────┼─────────────────────────┼────────────────┤
│ max_pooling2d (MaxPooling2D)      │ (None, 15, 15, 32)      │              0 │
├───────────────────────────────────┼─────────────────────────┼────────────────┤
│ conv2d_1 (Conv2D)                 │ (None, 13, 13, 64)      │         18,496 │
├───────────────────────────────────┼─────────────────────────┼────────────────┤
│ max_pooling2d_1 (MaxPooling2D)    │ (None, 6, 6, 64)        │              0 │
├───────────────────────────────────┼─────────────────────────┼────────────────┤
│ conv2d_2 (Conv2D)                 │ (None, 4, 4, 64)        │         36,928 │
└───────────────────────────────────┴─────────────────────────┴────────────────┘
 Total params: 56,320 (220.00 KB)
 Trainable params: 56,320 (220.00 KB)
 Non-trainable params: 0 (0.00 B)
```

Fill in the blanks:

- The first **Conv2D** layer extracts _features_ from images.
- An input image of size 32×32×3 means it has _32_ height, _32_ width, and _3_ color channels.
- **MaxPooling2D** helps by _reduce the spatial dimensions_ the feature map size, reducing computation.
- **Flatten()** converts the feature maps into _a 1D vector_ before feeding into the Dense layer.

Step 3 : Evaluating the Model

- What accuracy did the model achieve after training? _0.6986_
- Did the validation accuracy improve? _It improve at start and then it stop and decrease around the middle (May be overfitting)_
- Capture classification report



```
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 13ms/step
              precision    recall  f1-score   support

    airplane       0.59      0.85      0.70      1000
  automobile       0.86      0.78      0.82      1000
        bird       0.63      0.56      0.59      1000
         cat       0.50      0.54      0.52      1000
        deer       0.69      0.62      0.65      1000
         dog       0.63      0.59      0.61      1000
        frog       0.88      0.68      0.76      1000
       horse       0.70      0.80      0.75      1000
        ship       0.80      0.81      0.80      1000
       truck       0.84      0.76      0.80      1000

    accuracy                           0.70     10000
   macro avg       0.71      0.70      0.70     10000
weighted avg       0.71      0.70      0.70     10000
```