**WEEK2: CNN**

This notebook demostrates a **Convolutional Neural Network (CNN)** to classify images of cats and dogs. CNNs are particularly effective for image tasks because they can capture local spatial features (edges, textures, shapes) and gradually combine them into higher-level representations.

***Our CNN model consisted of:***

1. **First Convolutional Block**: Conv2D with 32 filters (3×3 kernel, ReLU activation), MaxPooling2D (2×2)
2. **Second Convolutional Block**: Conv2D with 64 filters (3×3 kernel, ReLU activation), MaxPooling2D (2×2)
3. **Fully Connected Layers**: Flatten, Dense layer with 128 neurons (ReLU), Output Dense layer with 1 neuron (Sigmoid activation)

The model was compiled with the **Adam optimizer, binary crossentropy loss**, and **accuracy** as the evaluation metric.

***Training procedure:***

* **Epochs:** 15 (with EarlyStopping and ModelCheckpoint)
* **Training accuracy:** ~**88.9%**
* **Validation accuracy:** ~**83.3%**
* **Training loss:** 0.27
* **Validation loss:** 0.43

From the accuracy & loss curves, we observed that the model began **overfitting after ~10 epochs.** Validation accuracy plateaued while training accuracy continued to improve.

***Evalution:***

* **Final Accuracy (Test Set):** ~**83%**
* **Confusion Matrix: [[1214 1286], [1197 1303]]** Correctly classified: 1214 cats, 1303 dogs, Misclassified: 1286 cats → dogs, 1197 dogs → cats, The model struggled almost equally with both classes but had slightly **more difficulty recognizing cats**.

This indicates that the model did not learn strong class-specific features and was essentially guessing (around random chance).

***Conclusion:***

This project demonstrated the full pipeline of building, training, and evaluating a CNN for binary image classification. While the model showed some potential (~83% validation accuracy), the test results revealed shortcomings, suggesting the need for deeper architectures, transfer learning, and stronger regularization.