Dogs vs Cats Image Classification using CNN

# **Project Overview**

**Image Processing**

Ziad Alaa Osman 202203811

Ramez Fawzy Ragheb 202203946

Ziad Waleed Waheed 202200703

Ahmed Naser Ahmed 202201563

Khaled Nasr Eldeen 202203490

Lab Instructor(s): Ahmed Khaled

Information Systems and Computer Science Faculty

October 6 University

May 12, 2025

# Brief Introduction

This project presents a Deep Learning Image Classifier built to distinguish between cats and dogs in digital images. Using a Convolutional Neural Network (CNN), the model learns abstract image patterns and features that differentiate these two classes. It is trained and evaluated on the popular Dogs vs Cats dataset, which is widely used for benchmarking image classification techniques.

Problem Definition

Accurately classifying animals in images is a common and practical challenge in computer vision. Manual classification is slow and error-prone, especially at scale. Traditional machine learning methods struggle with raw image data, and feature engineering is often insufficient. Hence, a deep learning solution that automatically learns features from data is more robust and scalable.

Figure 1: High-level data flow illustrating the full image classification pipeline.

# Solution Definition & Core Functionality Examples

The implemented CNN model performs the following:

* **Data Preprocessing**: Images are resized to 128x128 and normalized.
* **Model Training**: A CNN is built using Keras, with convolutional, pooling, and dense layers.
* **Prediction**: Given a new image, the model outputs a class label ("Cat" or "Dog").

|  |  |
| --- | --- |
| 1 | model = Sequential([ |
| 2 | Conv2D(32, (3,3), activation='relu', input\_shape=(128,128,3)), |
| 3 | MaxPooling2D(2,2), |
| 4 | Conv2D(64, (3,3), activation='relu'), |
| 5 | MaxPooling2D(2,2), |
| 6 | Flatten(), |
| 7 | Dense(128, activation='relu'), |
| 8 | Dense(1, activation='sigmoid') |
| 9  10 | ]) |

Listing 1: CNN Model Structure (Simplified)

|  |  |
| --- | --- |
| 1 | uploaded\_file = st.file\_uploader("Upload an image") |
| 2 | if uploaded\_file: |
| 3 | img = load\_img(uploaded\_file, target\_size=(128, 128)) |
| 4 | img\_array = np.array(img) / 255.0 |
| 5 | pred = model.predict(img\_array.reshape(1, 128, 128, 3)) |
| 6 | label = "Dog" if pred[0] > 0.5 else "Cat" |
| 7  8 | st.write("Prediction:", label) |

Listing 2: Image Upload and Prediction via Streamlit

# Code & Libraries Used

* **Language**: Python 3.x
* **Core Libraries**:
  + TensorFlow/Keras – Model creation and training
  + NumPy – Image preprocessing
  + Streamlit – Web-based user interface
  + Matplotlib – Image display and visualization
  + PIL – Image loading and processing