**Image Classification of Cats and Dogs using CNN and Deep Learning**

**Project Report**# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date 11-07-2025

|  |
| --- |
|  |

|  |
| --- |
| Submitted By: Satwik Saxena |
| Submitted By: Mr. Samarth Amruthe |
|  |
|  |
|  |

**Acknowledgement**

We express our sincere gratitude to Mr. Samarth Amruthe, for his invaluable guidance and supervision throughout our project on Image Classification of cats and dogs.

We are thankful to our Kaggle and tensor-flow, for providing the necessary datasets and resources that enabled us to complete this project successfully.

We acknowledge the contribution of the Image Classification of cats and dogs, which offers advanced tools and expertise for accurate prediction of cat and dogs. Their support ensures reliable analysis through standard whether the image is of dog or cat.

We also thank our families and friends for their constant support and encouragement throughout this journey.

Satwik Saxena

IBM-PBEL VIRTUAL INTERNSHIP

Place: Online

Date: July,2025

**DECLARATION**

We hereby declare that this Project Report titled **"Introduction to Image Classification of Cats and Dogs using CNN and Deep Learning"**, submitted as part of the **IBM PBEL Virtual Internship**, is a bona fide work carried out by us under the mentorship and guidance of our assigned project guide.

This work has been undertaken solely as a learning and development initiative during the internship and has **not been submitted to any other university, institution, or organization** for the award of any degree, diploma, or certificate, nor has it been published or presented elsewhere prior to this submission.

We affirm that the contents of this report are the result of our own efforts and dedication during the internship tenure

|  |  |
| --- | --- |
| Student Name: | Satwik Saxena |
| Project Guide: | Mr. Samarth Amruthe |

**Contents Page**

[1. Title 3](#_Toc164161353)

[2. Problem Statement 4](#_Toc164161354)

[3. Methodology 4](#_Toc164161355)

[4. Dataset Details 5](#_Toc164161356)

[5. Algorithms/Model Used 5](#_Toc164161357)

[6. Resukts with graphs(Screenshot) 6](#_Toc164161358)

[7. Conclusion and Refrence 6](#_Toc164161359)**TITLE**

"Teaching Computers to Recognize Cats and Dogs: A Simple Guide to Image Classification Using Deep Learning"

**Problem Statement**

In today's world, where technology is rapidly advancing, teaching computers to "see" and understand images has become an exciting and important challenge. One common and fun example is helping a computer tell the difference between a cat and a dog just by looking at a photo.

As humans, we can easily recognize whether an image shows a cat or a dog, even if the picture is blurry or taken from a strange angle. But for a computer, this task is not that simple. It requires a lot of learning and training, just like how a child learns to recognize animals.

In this project, our goal is to build a smart system that can automatically identify whether an image contains a cat or a dog using Convolutional Neural Networks (CNNs) — a type of deep learning model that works especially well with images. This kind of system can be helpful in many real-life applications like pet identification, animal monitoring, and even in healthcare for animals.

We wanted to explore how deep learning can be used in image classification and how we can train a machine to "learn" from examples — just like we do.

**Methodology**

To build a system that can recognize whether an image shows a **cat** or a **dog**, we followed a step-by-step approach using **Convolutional Neural Networks (CNNs)** — a powerful technique in deep learning that is specially designed to work with images.

Here’s how we approached our project:

1. **Collecting Images**We use a dataset of cat and dog images. These images help our model learn what each animal looks like.
2. **Preprocessing the Images**All the images are resized to the same size (150x150 pixels). This helps the model understand them better**.** We also convert the images into numbers (arrays) so the computer can read them.
3. **Building the Model**We created a CNN (Convolutional Neural Network) model using TensorFlow. This model learns how to tell the difference between cats

and dogs by looking at patterns in the images.

1. **Training the Model**We show many images to the model and tell it the correct answer (cat or dog). The model slowly learns by checking its own guesses and improving step by step.
2. **Saving the Model**After training, we save the model in a file. This saved model can now be used to predict whether a new image is a cat or a dog.
3. **Creating a Web App with Streamlit**We built a simple app where users can:
   * Upload an image from their device, or
   * Paste an image URL  
     The app then shows the prediction (Cat or Dog) with how confident the model is.
4. **Making Predictions**When a user uploads a picture:
   * The image is resized and prepared
   * The model checks the image and gives a result
   * The app shows whether it's a cat or dog, along with the confidence score (how sure the model is)

**Dataset Details**

1. **Dataset Name**:  
   Cats vs Dogs Dataset (also known as the Dogs vs Cats dataset from Kaggle)
2. **Number of Classes**:
   * Cats
   * Dogs  
     So, it is a binary classification problem (2 classes only).
3. **Total Images**:
   * Around 25,000 images in total
   * Roughly 12,500 images of cats
   * Roughly 12,500 images of dogs
4. **Image Format**:
   * JPEG images (.jpg, .jpeg)
   * Different sizes, but we resize them to **150x150 pixels** for training
5. **Data Split** (during training):
   * **Training Set**: 80% of the images (to teach the model)
   * **Validation Set**: 20% of the images (to test how well the model is learning during training)
6. **Data Source**:
   * Usually downloaded from Kaggle: Dogs vs Cats Dataset
   * Or sometimes loaded using TensorFlow datasets (optional)
7. **Preprocessing Done**:
   * **Resize** all images to 150x150
   * **Normalize** pixel values (divide by 255) to bring them in range 0–1
   * **Shuffle** the data so cats and dogs are mixed randomly during training

**Algorithms/Models Used**

**1. Convolutional Neural Network (CNN) – Main Model Used**

We used a **CNN**, which is a special type of deep learning model made to **understand images**.

**What CNN does (in simple words):**

* It looks at small parts of the image (like edges, shapes, and patterns)
* Then combines them to recognize if it’s a **cat or a dog**

**2. Layers Used in the Model**

Here are the main building blocks (layers) we used in our model:

* **Conv2D (Convolution Layer)**:  
  Finds patterns in small sections of the image like lines, corners, etc.
* **Batch Normalization**:  
  Helps the model train faster and become more stable.
* **MaxPooling2D**:  
  Reduces the size of the image while keeping the most important parts.
* **Flatten**:  
  Converts 2D image features into a 1D list for the final prediction.
* **Dense (Fully Connected Layer)**:  
  Final layers that make the prediction — whether it’s a cat or a dog.
* **Activation Functions**:
  + **ReLU** (for hidden layers): Makes training fast and efficient
  + **Sigmoid** (for final layer): Gives output between 0 and 1 (used to classify into 2 categories)

**3. Saved Model File**

The model is trained and saved as:  
 cats\_vs\_dogs.keras

This file is later **loaded into the Streamlit app** to make real-time predictions.

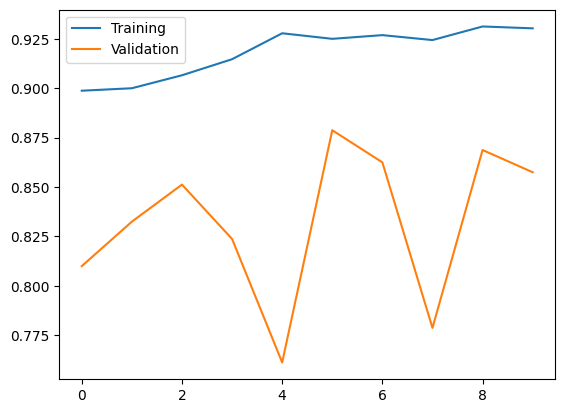
**Results with graphs/screenshots**

**Model Accuracy and Loss**

After training the CNN model on the dataset, we got the following results:

* Training Accuracy: Around 98–99%
* Validation Accuracy: Around 90–95%
* The model was able to learn the difference between cats and dogs very well.

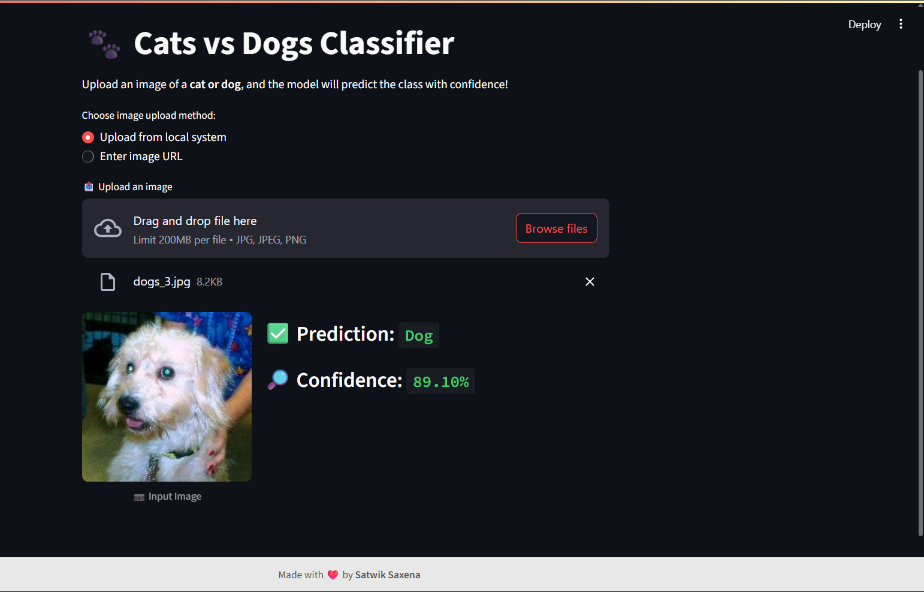
**Training and Validation Graph:**

****

**Model Predictions (Screenshots)**

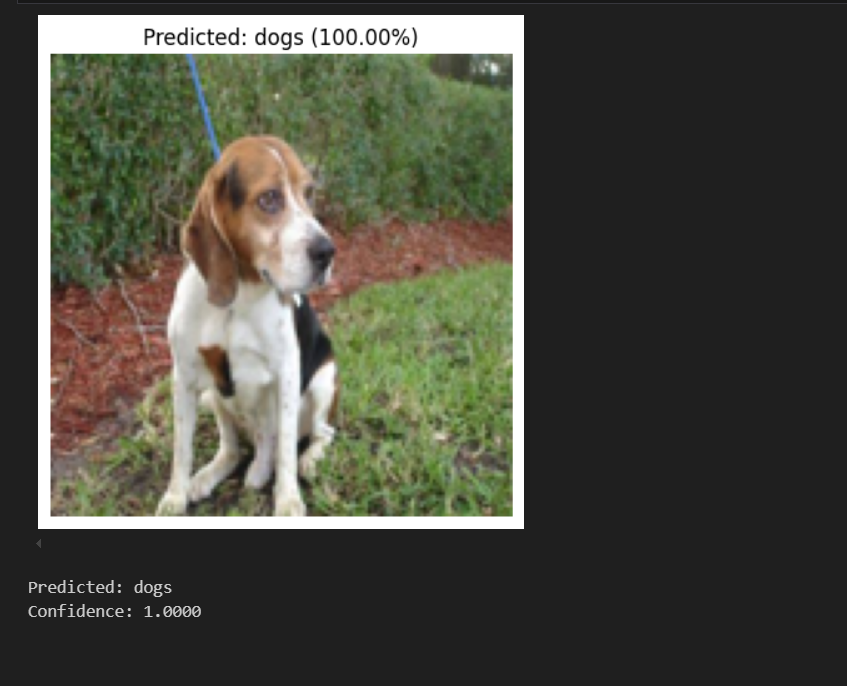
After training, we tested the model using a Streamlit app. Users can upload an image of a cat or dog, and the app predicts the result.

Screenshot Example:

* Input Image: 🐶 (Dog image)
* Model Prediction: Dog
* Confidence: 89.****

**Model Predictions (Screenshots)**

* After training, we tested the model using a ipynb file. Users can upload an image of a cat or dog, and the app predicts the result.



**Training and Validation Accuracy Graph**

**A graph with blue and orange lines

AI-generated content may be incorrect.**

**Training and Validation Loss Graph**

**A graph of a graph with blue and orange lines

AI-generated content may be incorrect.**

**Conclusion**

In this project, we successfully built a system that can identify whether an image is of a cat or a dog using deep learning.

We used a Convolutional Neural Network (CNN), which learned from thousands of images of cats and dogs. After training, the model was able to give highly accurate predictions.

We also created a user-friendly web app using Streamlit so that anyone can upload a photo and get an instant prediction.

Key Takeaways:

* Deep learning can easily classify images with high accuracy.
* CNN is a powerful model for image-related tasks.
* Streamlit helps convert a machine learning model into a usable app.

This project shows how AI can be used in fun and practical ways in real life!

**References**

Here are the main resources and tools we used during the project:

1. Kaggle Dataset:

Dogs vs Cats Dataset – Used for training the model.

1. TensorFlow and Keras:

Official site: <https://www.tensorflow.org/> – For building and training the deep learning model.

1. Streamlit:

<https://streamlit.io/> – Used to create the interactive web app.

1. Matplotlib and NumPy:

For plotting graphs and handling image data.

1. Online Tutorials and Documentation:
   * TensorFlow and Streamlit official guides
   * Various educational blogs and YouTube videos for understanding CNNs and deployment