**Doodle Draw**

Define the project scope:

1. Select Categories:

Here are 12 category suggestions that are both recognizable and relatively simple to draw:

* **Cat:** A common subject with distinctive features like whiskers and ears.
* **Dog:** Similar to cats but with a different outline and posture.
* **House:** Offers simple geometric shapes (squares, triangles for the roof) that are easy to sketch.
* **Tree:** Can be drawn with a trunk and a canopy; allows for variation in style.
* **Car:** A bit more complex but still recognizable with basic shapes.
* **Airplane:** Provides a fun challenge with wings and a fuselage.
* **Fish:** Simple curves and fins make for a clear and easy sketch.
* **Flower:** Often drawn as a circle with petals; versatile and stylized.
* **Cup:** A simple container shape that's familiar and easy to reproduce.
* **Clock:** Combines a circle with clock hands; a straightforward design.
* **Star:** Geometric and simple, yet visually distinct.
* **Umbrella:** An interesting shape with a curved canopy and handle.

1. Determine the Project Goals:

 **Interactive Web Drawing Interface:**

* Develop a straightforward HTML5 canvas for users to draw using a fixed brush.
* Include basic tools: drawing, an eraser, and a clear button.

 **Real-Time Recognition:**

* Integrate a simple CNN model to classify the drawing in real time.
* Display the predicted category along with a confidence score.

 **Responsive Design:**

* Ensure the application works smoothly across desktops, tablets, and smartphones.

 **Seamless Integration:**

* Connect the drawing interface with the model either via a backend API (e.g., Flask) or directly in the browser using TensorFlow.js.

1. Data Considerations:

 **Data Source & Selection:**

* **Quick Draw Subset:** Utilize a small subset of the Quick Draw dataset for your chosen categories. This provides a diverse range of user-generated sketches while keeping the volume manageable.
* **Custom Drawings:** Alternatively, or in addition, create your own drawings for each category to ensure the dataset aligns with your project’s simplicity.

 **Dataset Size:**

* Aim for approximately 200–300 examples per category. This should be sufficient to train a simple CNN without overwhelming the system.
* If needed, combine data sources to reach the desired number of examples per category.

 **Image Preprocessing:**

* **Uniform Dimensions:** Resize all images to a consistent size (e.g., 28×28 or 64×64 pixels).
* **Grayscale Conversion:** Convert sketches to grayscale, as color isn’t essential for simple drawing recognition.
* **Normalization:** Scale pixel values to a [0, 1] range to improve training efficiency.

 **Dataset Organization:**

* **Train-Test Split:** Partition your dataset into training, validation, and test sets (for example, 70% training, 15% validation, and 15% test) to reliably evaluate model performance.
* **Labeling:** Ensure each image is correctly labeled according to its category.

 **Data Augmentation (Optional):**

* Implement basic transformations such as slight rotations, shifts, or scaling. This can help your model generalize better, especially if the dataset is on the smaller side.

**2.1 Data Collection**

1. **Choose Your Data Source:**
   * **Quick Draw Subset:**  
     Download a small subset of the Quick Draw dataset for each of your selected categories. This dataset is readily available and provides a variety of sketches.
   * **Custom Drawings:**  
     Alternatively, you can generate your own sketches using a simple drawing tool. This may be useful if you want complete control over the style and quality of your data.
2. **Data Volume:**
   * Aim for **200–300 examples per category**. This amount should be sufficient to train a simple CNN while keeping the project manageable.
3. **Organize Your Data:**
   * **Folder Structure:**  
     Create a folder for each category (e.g., data/cat, data/dog, etc.) and store the corresponding images inside.
   * **Metadata (Optional):**  
     Maintain a simple CSV or JSON file mapping each image file to its label, which can be handy when loading and splitting your dataset.

**2.2 Data Preprocessing**

1. **Uniform Image Dimensions:**
   * **Resizing:**  
     Standardize the size of all images. Common choices are **28×28 pixels** (like MNIST) or **64×64 pixels**, depending on your desired level of detail.
   * **Tools:**  
     Use libraries such as Pillow, OpenCV, or TensorFlow’s image processing utilities to handle resizing.
2. **Grayscale Conversion:**
   * Convert images to grayscale since color information is generally unnecessary for recognizing simple sketches.
3. **Normalization:**
   * Normalize pixel values to the range **[0, 1]**. This step helps your model train faster and more reliably.
4. **Data Augmentation (Optional):**
   * **Techniques:**  
     Consider slight rotations, translations, or scaling to increase the diversity of your training data.
   * **Benefits:**  
     This can help the model generalize better, especially if your dataset is on the smaller side.
5. **Dataset Splitting:**
   * **Training, Validation, and Test Sets:**  
     Split your dataset, for example, into 70% for training, 15% for validation, and 15% for testing. This allows you to monitor the model’s performance and adjust hyperparameters without overfitting.

You can download data for a specific category from the Quick Draw dataset by accessing the dedicated file for that category. Here's how you can do it:

1. **Visit the Quick Draw Data Page:**  
   Go to Quick Draw Data to see all available categories. Each category has its own downloadable file.
2. **Find the Direct Download Link:**  
   For the numpy bitmap version, the file URLs follow this pattern:

pgsql

CopyEdit

https://storage.googleapis.com/quickdraw\_dataset/full/numpy\_bitmap/<category>.npy

For example, for the "cat" category, the URL is:

pgsql

CopyEdit

https://storage.googleapis.com/quickdraw\_dataset/full/numpy\_bitmap/cat.npy

1. **Download Programmatically (Python Example):**  
   You can download the file using Python’s urllib module. Here's a short script:

python

CopyEdit

import urllib.request

# Specify the category you're interested in

category = "cat"

url = f"https://storage.googleapis.com/quickdraw\_dataset/full/numpy\_bitmap/{category}.npy"

file\_name = f"{category}.npy"

# Download the file

urllib.request.urlretrieve(url, file\_name)

print(f"{file\_name} downloaded successfully!")

1. **Loading the Data:**  
   Once downloaded, you can load the numpy array using numpy.load:

python

CopyEdit

import numpy as np

data = np.load("cat.npy")

# Optionally, select a subset if the file is large:

sample\_data = data[:300] # Take first 300 samples

print("Shape of the data:", sample\_data.shape)

executing the above code in jupyter notebook.