**Form 3: Methodology**

1. **Team No: 2**
2. **Project Title: LSTM based OCR**
3. **Proposed Method:**

Data Collection

The dataset by Achanta and Hastie has 460 classes and 160 samples per class which made up 76000 images.

Each character has been augmented with 20 different

fonts downloaded from [13]. Using all the fonts for gutintham

variants and 3 fonts for vattu variants, all possible vattu and

gunintham forms of a character have been manually entered

in Microsoft Word. We then changed the font size from 15 to

40 with a step size of 5 covering 6 different font sizes for all

the variants of each character. We then took screen-shots of

each page containing these characters and used our segmen-

tation algorithm on them to get the individual characters.

Data Preprocessing:

* + Clean and preprocess the dataset, handling missing values and converting categorical features into a suitable format for machine learning.
  + The dataset has 17387 categories and nearly 560 samples per

class. All the images are of size 32 × 32. There are 6,757,044

training samples, 972,309 validation samples and 1,934,190

test samples which add upto 1 million images (10 GB).

Feature Engineering:

Settings for the sliding window, varying frame height between

20 and 60 pixels, frame width between 1 and 8 pixels, and

the overlap ratio between 0% and 75%. Modifying these

parameters allows one to adjust the computational cost of the

model at a tradeoff in accuracy.

Model Evaluation:

* + Evaluate the model's performance on the testing dataset using metrics such as accuracy.

Front-End Development:

* + Develop a user-friendly front-end using a web framework (e.g., Flask, Django, or any other of your preference).
  + Design an interface where users can input their image.

Model Integration:

* + Integrate the trained machine learning model into the front-end application.
  + Use the appropriate libraries or frameworks to make predictions based on the user-input.

Result Presentation:

* + Display the prediction result on the front-end, clearly indicating the input image in plain text.

Testing and Validation:

* + Testing our application thoroughly, both for functionality and security

1. **Proposed Method Illustration:**

A diagram of a software process

Description automatically generated

**Signature of the Supervisor**