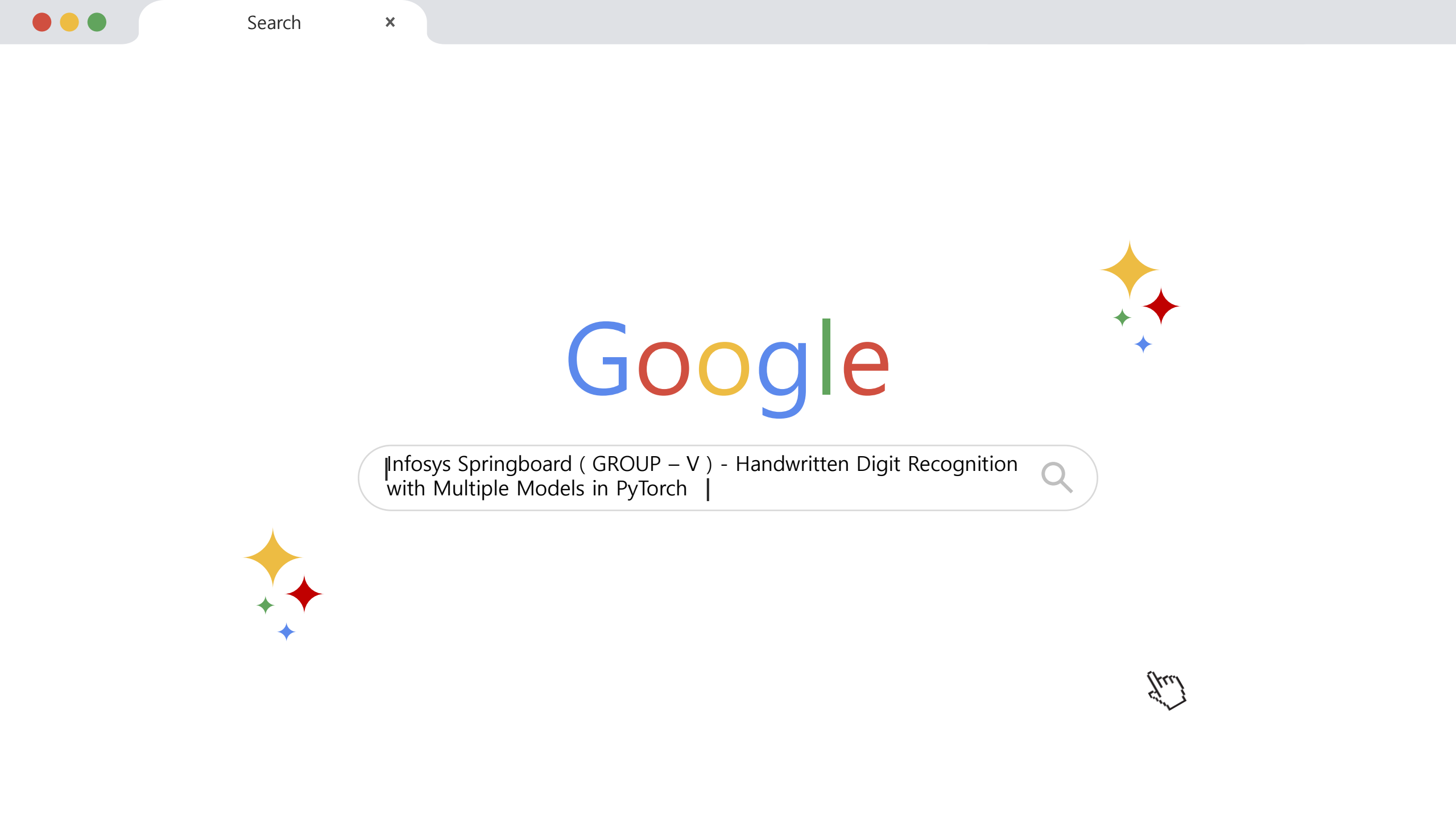

Search
Google
Infosys Springboard (GROUP – V) - Handwritten Digit Recognition with Multiple Models in PyTorch

Google Comprehensive Project Breakdown

Technologies

The project employs Python with Flask as the web framework, utilizing TensorFlow, Scikit-learn, matplotlib, and NumPy for machine learning and data visualization. Frontend technologies include.

Dataset

The MNIST dataset (Modified National Institute of Standards and Technology) is a large collection of handwritten digits that is commonly used for training various image processing systems.

Models

In order to create and test the accuracy and validation percentage , Group- V has implemented 3 different models for predicting the precise handwritten digits.

Workflow

The user draws a digit on the canvas using the mouse or touch input. The user then selects a model from the dropdown and clicks the "Predict" button. Canvas Image Data:

Search × Infosys Spring ×

https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown - Technologies

The project employs Python with Flask as the web framework, utilizing TensorFlow, Scikit-lear n, matplotlib, and NumPy for machine learning and data visualization. Frontend technologies include HTML, CSS, and JavaScript, with HTML5 Canvas API, Fetch API, and Chart.js for dyna mic data representation. Code is developed using Visual Studio Code and Jupyter Notebook.

- **Languages Used**: Python, Html, CSS, JavaScript
- **Framework:** Flask
- **Code Editors and Compilers:** Visual Studio Code, Jupyter Notebook
- **Libraries and API's: Python** – TensorFlow, Scikit-learn, matplotlib, NumPy
  JavaScript – HTML5 Canvas API, Fetch API, Chart.js

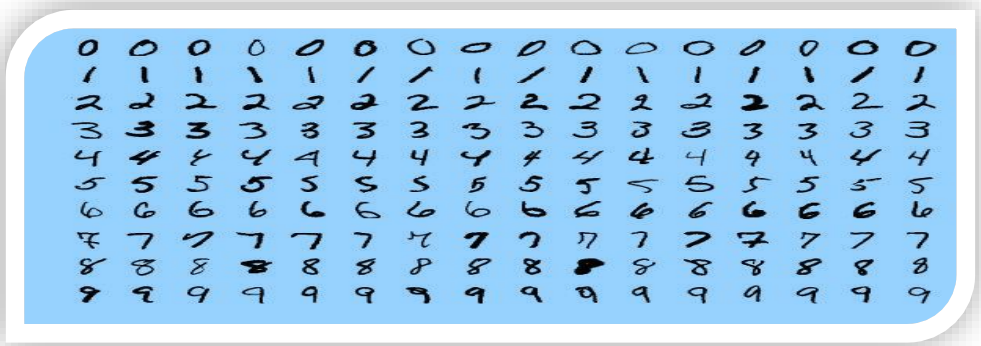
For extensive details and versions of all the required libraries requirements.txt file is given with the project

Google Comprehensive Project Breakdown - Dataset

- **MNIST Dataset**

The MNIST dataset (Modified National Institute of Standards and Technology) is a large col lection of handwritten digits that is commonly used for training various image processing systems

- **Content:** The dataset contains images of handwritten digits from 0 to 9.
- **Size:** It consists of 70,000 images in total, with 60,000 images for training
- **Image Details:** - Each image is a grayscale image of size 28x28 pixels.
  - Each pixel has a value ranging from 0 to 255, where 0 represents black and 255 represents white, with varying shades of grey in between.
- **Labelling:** Each image is labelled with the corresponding digit it represents (0 through 9)



For information and comparisons between different datasets please visit dataset.pdf uploaded on GitHub.

Search × Infosys Spring ×

https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown - Models

In order to create and test the accuracy and validation percentage , Group- V has implemented 3 different models for predicting the precise handwritten digits.

Such models are:

- LeNet-5
- Logistic Regression
- MLP(Multi-Layer Perceptron)

Detailed comparisons of results and accuracy have been documented into this file.

Google Comprehensive Project Breakdown – LeNet 5

LeNet is a convolutional neural network that Yann LeCun introduced in 1989.The LeNet-5 signifies CNN's emergence and outlines its core components. However, it was not popular at the time due to a lack of hardware, e specially GPU (Graphics Process Unit, a specialized electronic circuit designed to change memory to accelerate the creation of images during a buffer intended for output to a show device) and alternative algorithms, like SVM, which could perform effects similar to or even better than those of the LeNet.

Features of LeNet-5

- Every convolutional layer includes three parts: convolution, pooling, and nonlinear activation functions
- Using convolution to extract spatial features (Convolution was called receptive fields originally)
- **The average pooling layer** is used for subsampling.
- '**tanh**' is used as the activation function
- Using **Multi-Layered Perceptron** or **Fully Connected Layers** as the last classifier
- The sparse connection between layers reduces the complexity of computation

Google Comprehensive Project Breakdown – LeNet 5 Architecture

The generic architecture of LenNet-5 consists of multiple layers as described below:

- **Conv2D Layer**: 6 filters, 5x5 kernel, ReLU activation
- **MaxPooling2D Layer**: 2x2 pool size
- **Conv2D Layer**: 16 filters, 5x5 kernel, ReLU activation
- **MaxPooling2D Layer**: 2x2 pool size
- **Flatten Layer**: Flattening the 3D (4*4*16) outputs to 1D
- **Dense Layer**: 120 units, ReLU activation
- **Output Layer**: 10 units, softmax activation for multi-class classification

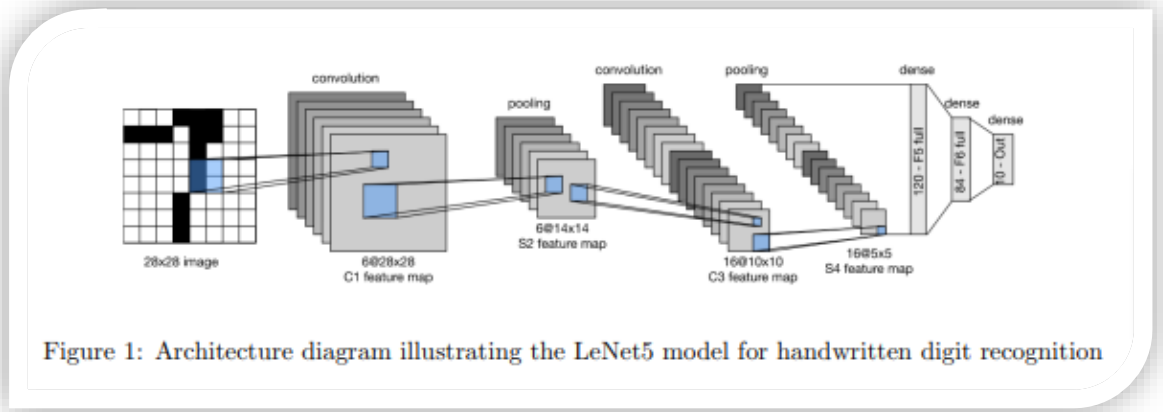


Figure 1: Architecture diagram illustrating the LeNet5 model for handwritten digit recognition

Our Lenet-5 model can be found in model.py file.

Search × Infosys Spring ×
https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown – LeNet-5 training and evaluation

Training Process:

- **Optimizer**: Adam
- **Loss Function**: Categorical cross entropy
- **Metrics**: Accuracy
- **Epochs**: 30
- **Batch Size**: 64

Training was performed for 10 epochs, with each epoch consisting of 938 iterations over the batch size of 64. The training process showed a steady decrease in loss and an increase in accuracy.

Evaluation Metrics:

- **Training Accuracy**: 99.54%
- **Validation Accuracy**: 98.92%
- **Test Accuracy**: 98.93%

The model's performance was evaluated using accuracy, where it achieved a high accuracy on both the training and test sets.

Inference Example

The trained model can predict the class of a single handwritten digit image. For example, given an input image of a digit '5', the model predicts with high confidence that the digit is '5'.

Search × Infosys Spring ×

https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown – LeNet 5 Conclusions

- The model performs exceptionally well on the MNIST dataset, achieving high accuracy (99%) on th e test set. For further improvements, consider increasing the number of epochs, experimenting with different network architectures, or using techniques like data augmentation to enhance the m odel's robustness.
- The CNN model excels in accuracy and is robust to overfitting, making it suitable for more comp lex datasets
- Consistent with training and validation accuracy, the CNN model also had the highest test accuracy

```
Test loss: 0.05117788165807724
Test accuracy: 0.9893000125885501
```

Search × Infosys Spring ×
https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown – Logistic Regression

Logistic regression is a supervised machine learning algorithm used for classification tasks which was originally developed and popularized primarily by Joseph Berkson, where the goal was to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors. This report provides an in-depth analysis of a logistic regression model used for digit classification on the MNIST dataset. Logistic regression is a simple yet effective model for multi-class classification tasks

Features of Logistic Regression:

- **Binary Outcome:** Logistic regression is used when the outcome variable is binary, meaning it has two possible outcomes.
- **Log-Odds Transformation:** It models the relationship between predictor variables and the log-odds of the outcome, mapping probabilities onto the entire real number range.
- **Linear Relationship:** Despite being a regression model, logistic regression establishes a linear relationship between predictors and the log-odds of the outcome.
- **Probabilistic Predictions:** Instead of predicting a binary outcome directly, it predicts the probability that an instance belongs to a particular class.

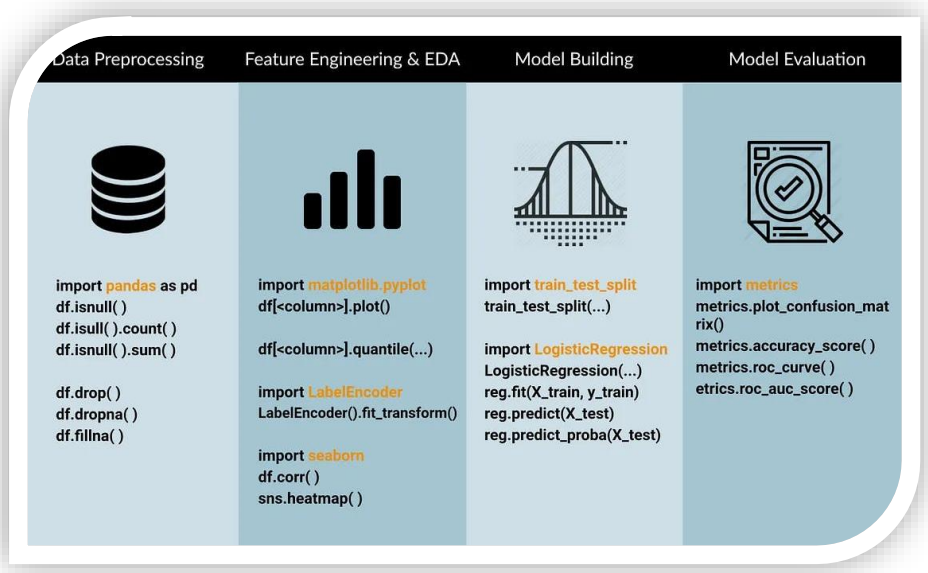
Search | Infosys Spring

https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown – Logistic Regression Architecture

The generic architecture of Logistic Regression consists of single layer as described below:

The logistic regression model was implemented using TensorFlow's Keras API. The model consists of a single dense layer with a SoftMax activation function to output probabilities for each of the 10 classes.



•**Single Layer**:

- Dense layer with 10 units and SoftMax activation function.

•**Model Summary**:

- Total parameters: 7,850
- Trainable parameters: 7,850
- Non-trainable parameters: 0

Our Lenet-5 model can be found in model1.py file.

Search × Infosys Spring ×
https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di
Google Comprehensive Project Breakdown – Logistic Regression training and e

Training Process:

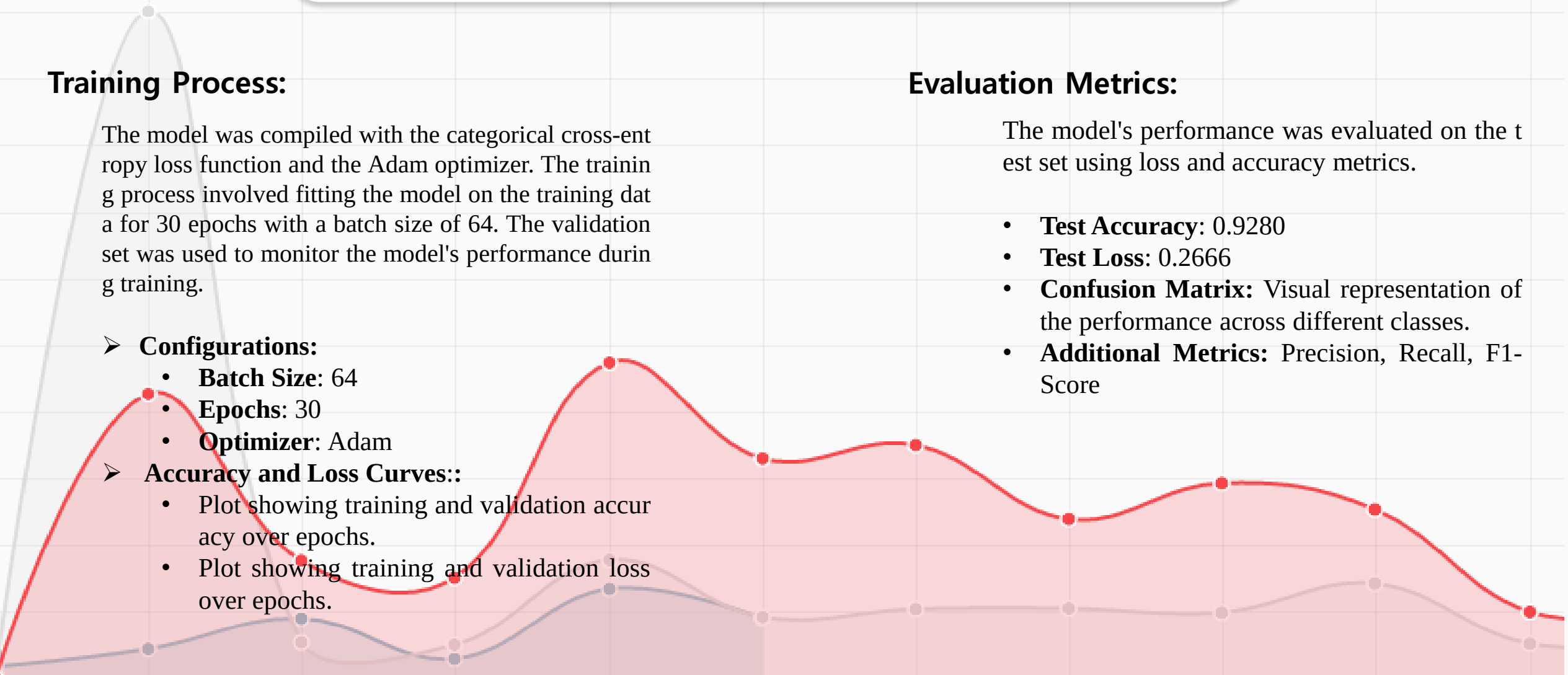
The model was compiled with the categorical cross-entropy loss function and the Adam optimizer. The training process involved fitting the model on the training data for 30 epochs with a batch size of 64. The validation set was used to monitor the model's performance during training.

- **Configurations:**
  - **Batch Size**: 64
  - **Epochs**: 30
  - **Optimizer**: Adam
- **Accuracy and Loss Curves::**
  - Plot showing training and validation accuracy over epochs.
  - Plot showing training and validation loss over epochs.

Evaluation Metrics:

The model's performance was evaluated on the test set using loss and accuracy metrics.

- **Test Accuracy**: 0.9280
- **Test Loss**: 0.2666
- **Confusion Matrix:** Visual representation of the performance across different classes.
- **Additional Metrics:** Precision, Recall, F1-Score



Search × Infosys Spring ×
https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown – Logistic Regression Conclusions

The logistic regression model achieved a test accuracy of approximately 92.8% on the MNIST dataset. This demonstrates that even a simple model like logistic regression can be effective for digit classification tasks. However, more complex mode ls such as Convolutional Neural Networks (CNNs) may achieve higher accuracy and better performance on more complex datasets.

Recommendations:

- **Model Complexity**: For higher accuracy, consider using more complex models like CNNs.
- **Data Augmentation**: Use data augmentation techniques to artificially increase the size of the training set.
- **Hyperparameter Tuning**: Experiment with different optimizers, learning rates, and batch sizes to further improve performance.
- **Regularization**: Implement regularization techniques like dropout or L2 regularization to prevent overfitting.

```
Test loss: 0.26662120223045 35
Test accuracy: 0.92799997329711 91
```

Google Comprehensive Project Breakdown – MLP (Multi-Layer Perceptron)

Multi-layer perceptron is also known as MLP discovered by Frank Rosenblatt. It is fully connected dense layers, which transform a ny input dimension to the desired dimension. A multi-layer perceptron is a neural network that has multiple layers. To create a neural network we combine neurons together so that the outputs of some neurons are inputs of other neurons. A multi-layer perceptron has one input layer and for each input, there is one neuron(or node), it has one output layer with a single node for each output and it can have any number of hidden layers and each hidden layer can have any number of nodes

Features of Multi-Layer Perceptron :

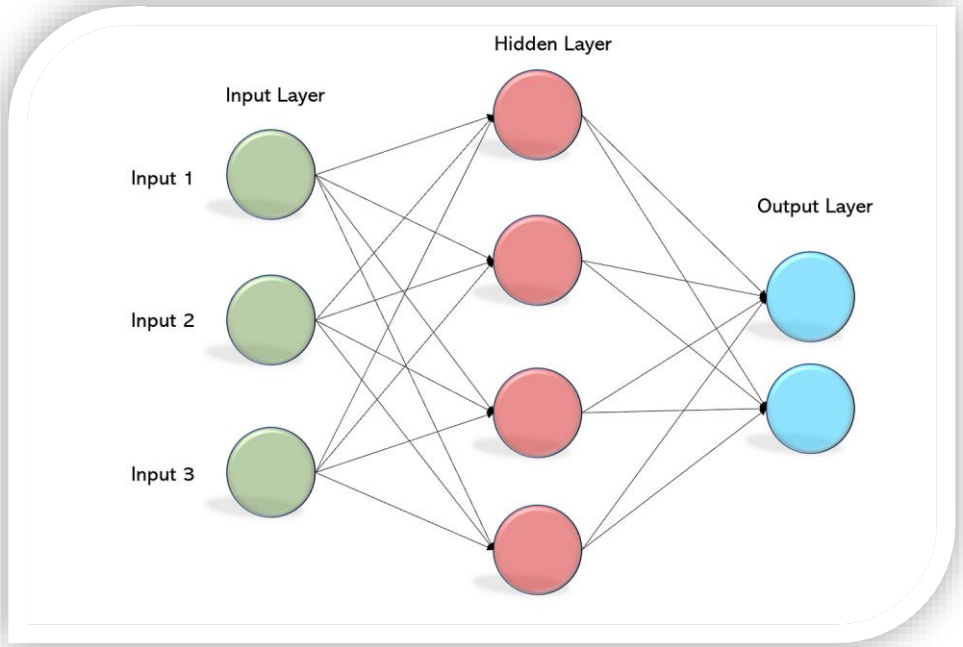
- **Layered Structure:** MLPs consist of multiple layers of neurons, including input, hidden, and output layers.
- **Non-linear Activation:** Neurons use non-linear activation functions (e.g., sigmoid, tanh, ReLU) to enable complex learning and decision-making.
- **Backpropagation:** Trained using backpropagation, adjusting weights iteratively to minimize prediction errors based on known outcomes.
- **Universal Approximators:** Capable of approximating complex functions when sufficiently large and appropriately structured.

Google Comprehensive Project Breakdown – MLP Architecture

The generic architecture of Multi Layer Perceptron consists of multiple layers as described below:

- **Layers**:
  - **Dense**: Fully connected layers with ReLU activation.
  - **Output Layer**: Dense layer with 10 units and softmax activation.

- **Model Summary**:
  - Total parameters: 399,210
  - Trainable parameters: 399,210
  - Non-trainable parameters: 0



Our Lenet-5 model can be found in model2.py file.

Search × Infosys Spring ×
https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di
Google Comprehensive Project Breakdown – MLP training and evaluation

Training Process:

Training was performed for 30 epochs, with each epoch consisting of 938 iterations over the batch size of 64. Th e training process showed a significant reduction in loss and increase in accuracy over epochs.

- **Configurations**:
- **Batch Size**: 128
- **Epochs**: 30
- **Optimizer**: Adam
- **Accuracy and Loss Curves**::
  - Plot showing training and validation accuracy over epochs.
  - Plot showing training and validation loss over epochs.

Evaluation Metrics:

- **Training Accuracy**: 99.80%
- **Validation Accuracy**: 97.67%
- **Test Accuracy**: 97.67%

The model's performance was evaluated using accur acy, where it achieved high accuracy on both the trai ning and test sets.

Inference Example :

The trained model can predict the class of a single h andwritten digit image. For example, given an input image of a digit '3', the model predicts with high con fidence that the digit is '3'.

Search × Infosys Spring ×
https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown – MLP Conclusions

- **Summary**:
  - MLP achieved high accuracy but is slightly less effective than CNN.
- **Strengths**:
  - Good at capturing non-linear patterns.
- **Weaknesses**:
  - Requires careful tuning of hyperparameters and architecture.
- **Improvements**:
  - Experiment with deeper networks and different activation functions.

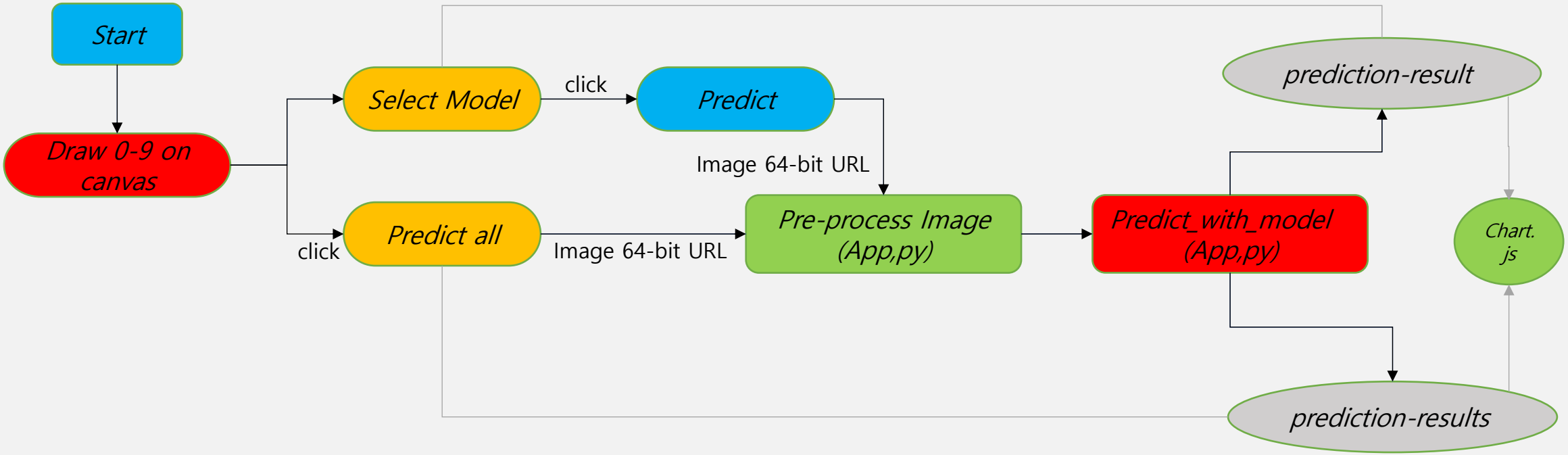
```
Test loss: 0.14988738298416138
Test accuracy: 0.9767000079154968
```

Search × Infosys Spring ×

https://www.google.com/search?q=Infosys+Springboard+(+GROUP+%E2%80%93+V+)++Handwritten+Di

Google Comprehensive Project Breakdown - Workflow

Data Flow Diagram – Handwritten Digit Recognition:



For More Detailed Workflow please visit here.

About Us:

This Project has been a collaborative effort from each member. Without any of us It would not been possible to complete the tasks on time.
We divided the tasks in between every team members. So the approach was quite simple to do everything as a team. The tasks done by each of us is listed below:

Rudraksh Gupta	Sneha Kushwaha	Yash Nilesh Shah
• Helped create user interface using flask and JavaScript. • Contribution in programming all three models • Maintenance of GitHub. • Beautified whole code. • Contributed in development of ppt. • Attended group meetings regularly.	• Programming all three models. • Designed flow charts to ensure clear project structure & execution. • Created detailed Model comparison report. • Tracked Progress on GitHub • Contributed in development of ppt. • Attended group meetings regularly.	• Contribution in the creation of GitHub repository. • Added all of us as collaborators.

Move Forward to User Interface

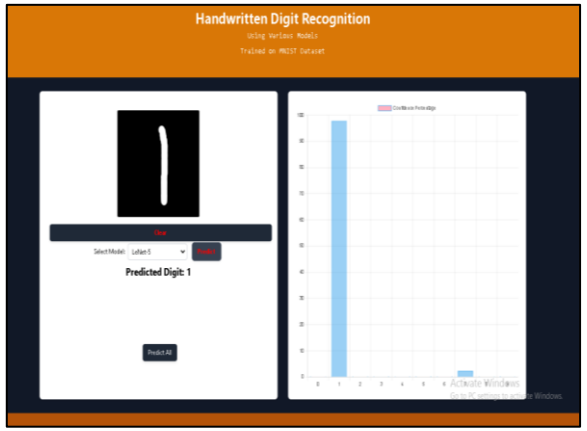
Search × Infosys Spring × UI ×

https://www.google.com/search?q=Handwritten+digit+Recognition+user+interface&rlz=1C1CHBF_enIN1

Google User Interface

Index Page

We start our project using flask virtual environment (project is yet to be deployed on the web). The above stated UI is seen

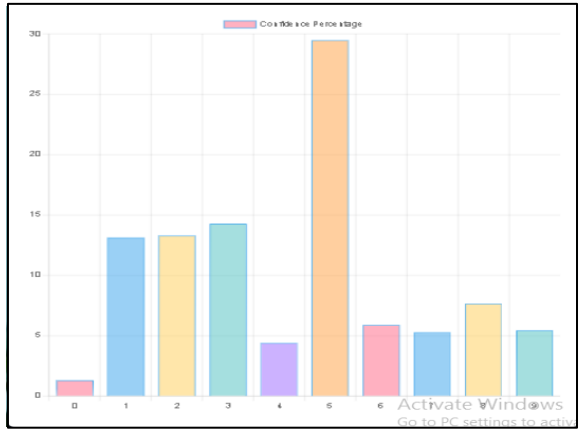


Prediction of Selected Model

In our UI, there is an option of selecting any model out of three for predicting the hand drawn digit.

Prediction of all Models

Moreover, users can all check the predictions of all models (lenet5, logistic regression and MLP) by a single click on “predict all” button.



Confidence Percentage

Every time the user clicks on predict button for predicting the digit. Confidence percentages are shown , which states that how much percent does the model has the confidence that result is correct.

Search | Infosys Spring | UI | Acknowledgement

https://www.google.com/search?q=Handwritten+digit+Recognition+user+interface&rlz=1C1CHBF_enIN1

Google

Thanks

Acknowledgement

Members of Group V are extremely delighted to work with our mentors Mr. Narendra Kumar and Mr. Nishant. The guidance provided by them were truly valuable. Our knowledge of artificial intelligence has surely been broaden by their Valuable insights. We would also like to oblige Infosys springboard for their impressive internship experience.

Search | Infosys Spring | UI | Acknowledgement

https://www.google.com/search?q=Handwritten+digit+Recognition+user+interface&rlz=1C1CHBF_enIN1

Google

Thanks

Thank you