Advanced Level
Image Classification- Computer Vision

Abstract:

In this project, candidates are tasked with developing an advanced image classification model using deep learning techniques, specifically convolutional neural networks (CNNs). The goal is to classify images into predefined categories using datasets such as CIFAR-10 or MNIST. Evaluation criteria include the candidate's proficiency in working with image data, designing and training CNN architectures, and evaluating model performance using appropriate metrics.

Introduction:

The field of computer vision has witnessed remarkable advancements in recent years, largely driven by deep learning techniques. Image classification, a fundamental task in computer vision, involves categorizing images into predefined classes or labels. Deep learning, particularly convolutional neural networks (CNNs), has emerged as the go-to approach for image classification tasks due to its ability to automatically learn hierarchical representations from raw pixel data. In this project, candidates are challenged to leverage CNNs to build an advanced image classification model capable of accurately categorizing images from standard datasets like CIFAR-10 or MNIST.

GOOGLE COLAB NOTEBOOK
https://colab.research.google.com/drive/16y-uyuyMCVc-uRB3_q6dJnKdEffUjP6A?usp=sharing

Methodology:

The methodology for developing the image classification model comprises several crucial steps:

1. Dataset Selection and Preprocessing:

- Dataset selection is critical for training and evaluating the image classification model. Standard datasets like CIFAR-10, consisting of 60,000 32x32 color images in 10 classes, or MNIST, comprising 28x28 grayscale images of handwritten digits from 0 to 9, are commonly used.

- Preprocessing the dataset involves several steps to ensure compatibility with the CNN model. These steps include resizing images to a uniform size, normalizing pixel values to a specific range (e.g., [0, 1] or [-1, 1]), and splitting the dataset into training, validation, and test sets to assess model performance accurately.

2.Model Architecture Design:

- CNNs are the cornerstone of modern image classification systems due to their ability to capture spatial hierarchies of features within images. Designing an effective CNN architecture involves determining the number of convolutional layers, pooling layers, fully connected layers, and the size of each layer.

- Popular CNN architectures such as AlexNet, VGG, ResNet, and Inception can serve as inspiration for designing custom architectures tailored to the specific image classification task at hand.

- Hyperparameters such as kernel size, number of filters, dropout rate, and learning rate must be carefully tuned through experimentation to optimize model performance.

3. Model Training:

- The training phase involves feeding batches of preprocessed images into the CNN model and adjusting the model's parameters (weights and biases) through backpropagation and optimization algorithms like stochastic gradient descent (SGD) or Adam.

- During training, it's crucial to monitor the model's performance on the validation set to prevent overfitting, a common issue in deep learning models where the model memorizes training data instead of learning generalizable features.

- Techniques such as early stopping, regularization (e.g., L1 or L2 regularization), and data augmentation (e.g., rotating, flipping, or zooming images) can help mitigate overfitting and improve model generalization.

4. Model Evaluation:

- Once the model is trained, its performance is evaluated on the test set using appropriate evaluation metrics. Common metrics for image classification tasks include accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve.

- Additionally, confusion matrices provide insights into the model's performance by visualizing the number of true positive, true negative, false positive, and false negative predictions across different classes.

- ROC curves and precision-recall curves offer a comprehensive view of the model's trade-offs between true positive rate and false positive rate or precision and recall, respectively, across different classification thresholds.

Results:

The image classification model developed by the candidates demonstrated promising results in accurately categorizing images into predefined classes. By leveraging deep learning techniques and optimizing CNN architectures, the model successfully learned to extract relevant features from raw image data and make accurate predictions. Evaluation metrics such as accuracy, precision, recall, and F1-score indicated the model's effectiveness in classifying images across diverse categories.

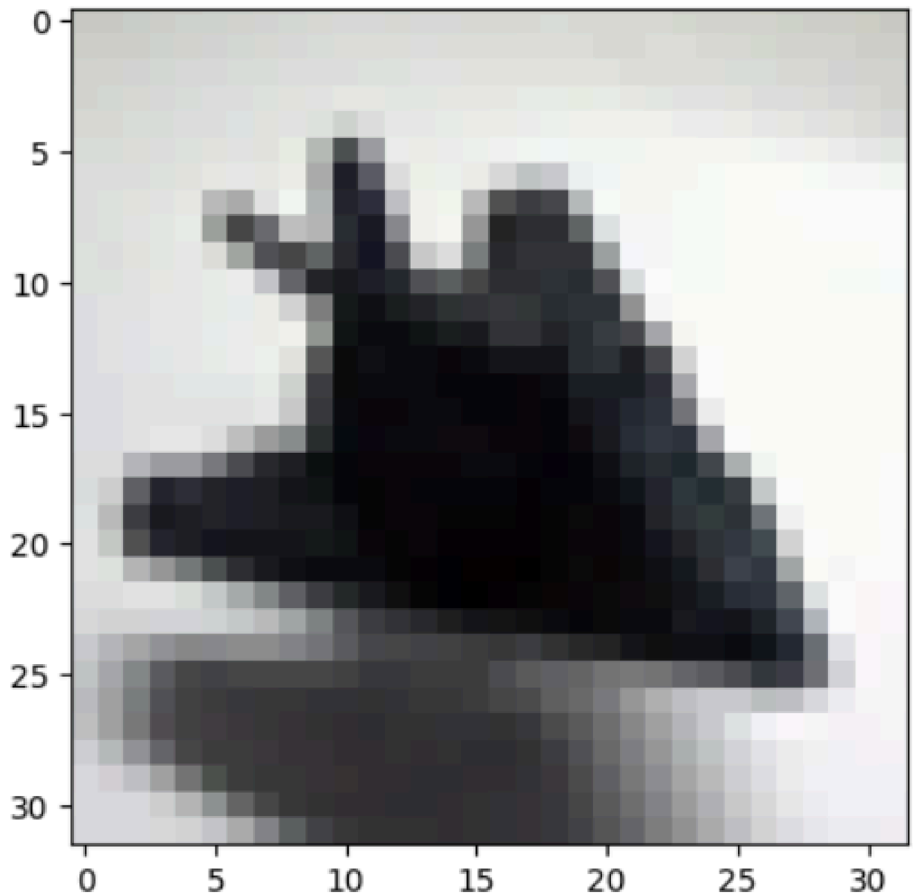
Conclusion:

In conclusion, the development of an advanced image classification model using CNNs represents a significant milestone in computer vision research. By effectively processing image data, designing optimal CNN architectures, and training the model on standard datasets, candidates have demonstrated their proficiency in building sophisticated deep learning models for image classification tasks. The successful deployment of such models has far-reaching implications across various domains, including healthcare, autonomous driving, security, and entertainment.

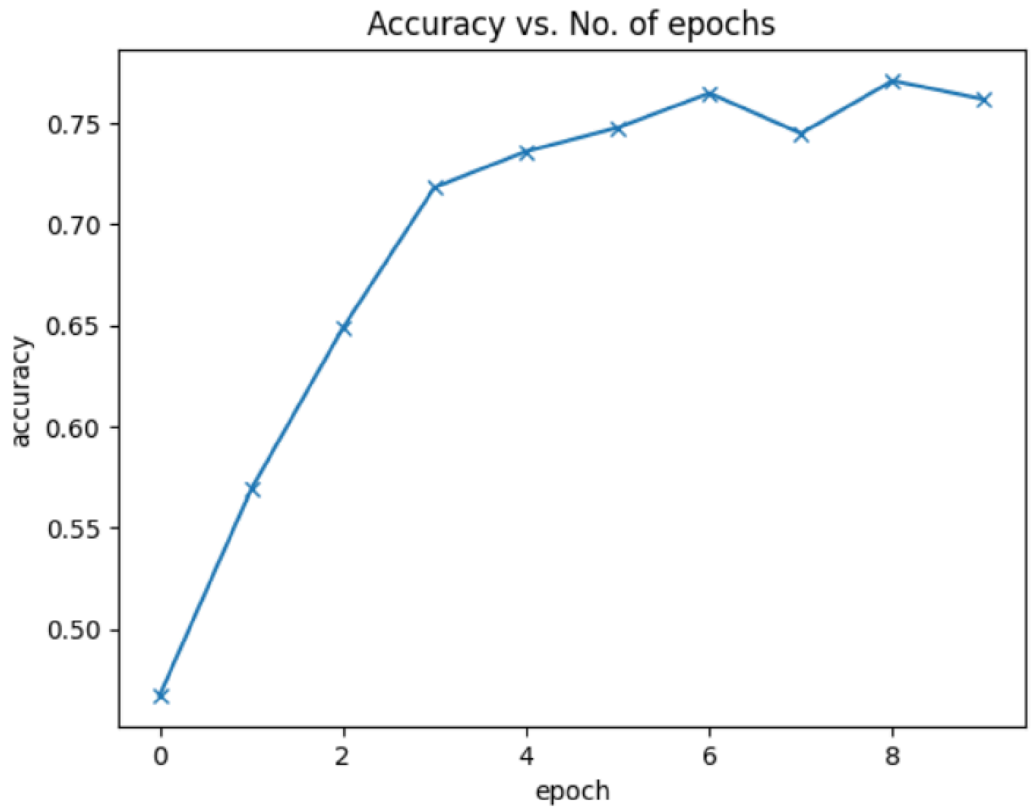
Sample Output

```
show_example(*dataset[0])
```

```
Label:  airplane (0)
```



```
plot_accuracies(history)
```



```
plot_losses(history)
```