Ice Cream VS Pizza Image Classifier (Scratch)

Introduction:

The primary goal of this project is to leverage Convolution Neural Networks (CNNs) implemented using the Keras framework to develop a robust model capable of distinguishing between images of Ice Cream and Pizza. Image classification is a fascinating field within computer vision, and our focus here is on creating an intelligent system that can automatically categorize images of these two distinct food items.

In the vast realm of computer vision, the task of image classification involves training a model to recognize and categorize objects within images. In our specific case, we want to teach our model to differentiate between two detectable treats: Ice cream and pizza. This distinction is not only visually interesting but also showcases the capability of deep learning to discern between intricate patterns and features within images.

Data Collection:

I collected the dataset from kaggle, comprising images of Ice Cream and Pizza.
Dataset Link: https://www.kaggle.com/datasets/hemendrasr/pizza-vs-ice-cream

Data Preprocessing:

- **Rescaling:** In simpler terms, rescaling ensures that the pixel values, which represent the intensity of colors in the images, are adjusted to a standardized range. This helps the neural network learn more efficiently and avoids potential challenges related to differences in input scales.
- **Data Augmentation:** To enhance the model's ability to generalize, augmentation techniques such as rotation and horizontal/vertical flips were applied during training. This artificially increases the diversity of the training dataset, helping the model better handle variations in the test set.
- **ImageDataGenerator for sets:** The Keras 'ImageDataGenerator' class was employed to efficiently create training, testing and validation sets. This class streamlines the process of loading and preprocessing images in batches, allowing for effective utilization of the available data during the training process.

Model Building (Scratch):

- **Purpose of Model Building:** In our project, the primary goal of building the neural network model is to create an intelligent classifier capable of distinguishing between images of Ice Cream and Pizza. The inherent complexity of visual patterns in these images demands a specialized approach, and Convolutional Neural Network (CNNs) emerge as a powerful solution.
  - ✓ Convolutional Neural Networks have demonstrated remarkable effectiveness in image classification tasks. Unlike traditional neural networks, CNNs are specifically designed to comprehend spatial hierarchies and local patterns within images. This makes them exceptionally well-suited for tasks where the arrangement of pixels contains crucial information. In our context, a CNN allows our model to automatically learn and extract features such as crust texture, toppings arrangement, or the unique patterns associated with ice cream cones.
  - ✓ We leverage the high-level neural networks API, Keras, for model construction. Keras simplifies the implementation of neural networks, providing an intuitive interface for building, training, and deploying models. Its user-friendly design allows us to focus on the architecture and configuration of our CNN without delving into low-level details.
- **Model Architecture Details:** Using Keras, we construct CNN architecture to process pixel information from ice cream and pizza images. The network is designed to recognize distinctive features through various types of layers:
  - ✓ **Convolutional Layers:** Convolutional layers play a key role in capturing local patterns and features. These layers convolve filters across the input image, allowing the network to automatically learn visual characteristics such as edges and textures. In this project the model takes with a shape of (417, 626, and 3) as input.
  - ✓ **Pooling Layers:** Pooling layers are employed for down sampling, reducing the spatial dimensions of the data. This helps in retaining important information while discarding less relevant details, contributing to computational efficiency. Two sets of Conv2D and MaxPooling2D layers were employed to extract features from the images.
  - ✓ **Fully connected Layers:** Fully connected layers are responsible for high-level reasoning. They take the features learned by the convolutional layers and make decisions based on the extracted representations.

- ✓ **Training on a Diverse Dataset:** Our model is trained on a diverse dataset comprising various images of both ice cream and pizza.
- ✓ **Project Significance:** This project holds significance not only for its exploration of the technical intricacies involved in implementing a Convoltional Neural Network (CNN) with Keras but also for the commitment to building the model entirely from scratch. By crafting the code from the ground up, we gain a deeper understanding of the underlying mechanisms and architecture of the neural network.
  - This endeavor is a journey into comprehending how neural networks can be meticulously designed and trained to make accurate decisions based on visual content.

Model Compilation:

During the model compilation phase, the neural network model has configured to optimize its performance using specific loss functions, optimizers, and evaluation metrics.

1. **Loss Function- Binary Crossentropy:** The choice of Binary Crossentropy as the loss function is well-suited for our binary classification task of distinguishing between ice cream and pizza. This loss function measures the dissimilarity between the actual labels, providing an effective signal for the model to minimize.
2. **Optimizer- Adam:** The Adam optimizer was employed to efficiently adjust the model's weights during training. Adam adapts the learning rates for each parameter individually, combining the advantages of both AdaGrad and RMSProp. This dynamic adaptation contributes to faster convergence and improved generalization performance.
3. **Metrics- Accuracy:** Model performance was assessed using accuracy as the primary metric. Accuracy represents the ratio of correctly predicted sample to the total number of samples, providing a straightforward measure of the model's overall correctness in its predictions.

This thoughtful configuration during model compilation sets the stage for effective training, ensuring that the model optimally learns from the dataset and makes accurate predictions on unseen data.

Training:

1. **Number of Epochs:** The model was trained for 20 epochs, iterating over the entire training dataset 20 times. This allows the model to learn from the data and adjust its parameters multiple times for improved performance.

2. **Steps per Epoch:** To efficiently utilize the training data, it was divided into batches, each containing 32 samples. With 718//32 batches processed per epoch, this approach ensures that the model undergoes frequent updates based on a diverse set of training examples.
3. **Validation Data:** The model's performance was assessed using a separate validation set 208//32 steps during each epoch. This step is crucial for evaluating the model's ability to generalize to unseen data and preventing over fitting.
4. **Callbacks:** Several callbacks were integrated to enhance the training process:
    - **Early Stopping:** This callback monitors the validation loss and halts training if it does not improve significantly, preventing over fitting.
    - **ModelCheckpoint:** It saves the model after each epoch only if the validation loss improves, ensuring that the best-performing model is retained.
    - **CSVLogger:** This callback logs training and validation metrics into a CSV file, allowing for easy tracking and analysis of the model's performance over epochs.

Model Evaluation:

After training our Convolutional Neural Network (CNN) model, we assessed its performance on the test set to gauge its effectiveness in distinguishing between ice cream and pizza.

- **Accuracy on test set:** The model exhibited an impressive accuracy of 61% on test set. This metric reflects the percentage of correctly classified instances, providing a robust measure of the model's overall performance in real-world scenarios.
- **Confusion Matrix:** In addition to accuracy, a confusion matrix provides a detailed breakdown of the model's predictions, showcasing true positives, true negatives, false positives, and false negatives. This matrix, if desired, offers deeper insights into the specific types of errors made by the model.

This evaluation phase allows us to validate the model's ability to generalize and make accurate predictions on new, unseen images. The achieved accuracy score serves as a quantitative measure of the model's success in distinguishing between the classes of internet.

Conclusion:

- **Achieved Accuracy:** The model has demonstrated commendable performance, achieving an accuracy score of 61% on the test set. This validates the effectiveness of the designed Convolutional Neural Network (CNN) in accurately classifying images of ice cream and pizza.

- **<u>Challenges Faced:</u>** Throughout the project, we encountered and successfully navigated various challenges. These included fine-tuning the model architecture for optimal performance, addressing data augmentation complexities, and ensuring efficient convergence during training.