Classification of Images (Pizza and Not Pizza) Using Machine Learning

Poojeetha Madugula

School of Computing, Wichita State University

Classification of Images (Pizza and Not Pizza) Using Machine Learning

In this paper we are presenting an observation of an image classification model designed in our project to differentiate between images of pizzas and images that do not contain pizzas. The project includes the whole data science life cycle where it started with the Data collecting and Data preprocessing and building a Convolutional Neural Network (CNN) model and finding out the model's performance. This paper describes the steps involved in the project, including data collection, data preprocessing, model development, and evaluation. This research goal is to contribute to the field of image classification and computer vision by understanding and developing the useful binary classification model. The final aim of the project is to fetch and get the high accuracy to distinguish between pizza and non-pizza images.

In this project, we focus on the task of distinguishing between images of pizzas and images that do not represent pizzas. The "Pizza and Not Pizza" project aims to build an image classification model which automatically identifies if a given image is a pizza or not.

Literature Review

Food Image Recognition by Using Convolutional Neural Networks (CNNs)

According to the research the study revealed the use of computer vision for visual object recognition, especially in the context of food image recognition. In the Food Image Recognition, they created a dataset with 5822 images in ten categories and employed a five-layer CNN for image classification. Initially, tested the bag-of-features (BoF) model with support vector machine (SVM), by performing 56% accuracy. However, the CNN model outperformed it notably, reaching a 74% accuracy. In this paper they used data augmentation techniques involving geometric transformations to increase the training dataset size, fetching an accuracy exceeding 90%. In the original data CNN has observed overfitting issues. In the research paper the authors suggested that further improvement could be achieved by expanding the image dataset, optimizing the network architecture, and adjusting hyperparameters.

Multi-classification of pizza using computer vision and support vector machine

According to this research paper our study is subjected to classifying pizza base, sauce spread, and the other toppings which can be prone to human errors due to its unpredictable behavior. To mention the image processing techniques combined with machine learning offer a clearer and steady approach. In this paper we observed that the Support Vector Machine (SVM) stands out for this multi-classification task. By using selected features as input, both one-versus-one and directed acyclic graph (DAG) methods achieved high multi-classification accuracy, with 89.17% and 88.33% for pizza base, 87.5% for pizza sauce spread, and 80.83% and 80.00% for pizza topping, respectively. The results indicate the prospect of computer vision systems in automating the classification of pizza base, sauce spread, and toppings.

Comparison of three methods for classification of pizza topping using different color space transformations

From this research paper, the study evaluated five different color space transformations (NRGB, HSV, I1I2I3, Lab*, and YCbCr) for classifying pizza toppings. In the research paper, they compared the performance of three SVM classifiers namely linear, polynomial, and RBF and two traditional classification methods (C4.5 and RBF_NN). When using Lab* the C4.5 classifier achieved the highest accuracy of 93.3%, while the RBF_NN classifier reached 86.7% accuracy with YCbCr, HSV, or Lab*. Among the SVM classifiers, the SVM classifier performed best with 96.7% accuracy when using the HSV color space transformation and has proved to be a successful approach for pizza topping classification using computer vision.

Inputs

The main goal of the project is to classify the images of food items into Pizza and not a Pizza with wood accuracy.

Schedule/ Timelines

Table 1

Tentative Schedule

Week Number	Work
Week 1	Topic Selection
Week 2	Topic Selection
Week 3	Data Acquisition
Week 4	Data Pre-Processing
Week 5	Data Pre-Processing
Week 6	Model Building
Week 7	Model Building
Week 8	Model Building
Week 9	Mid Term Presentation - Feedback
Week 10	Mid Term Presentation
Week 11	Model Building
Week 12	Model Building
Week 13	Model Building
Week 14	Presentation Preparation
Week 15	Final Presentation – Feedback

Week Number	Work
Week 16	Final Presentation

Note: The above table is based on our works from week -1.

Table 2

Member Roles

Names	Contributions
Poojeetha	Presentation, Model Building, Data Augmentation, Report
Vineeth	Topic Selection, Data Acquisition, Model Building, Report

Note: The above table is based on our work contributed for the project.

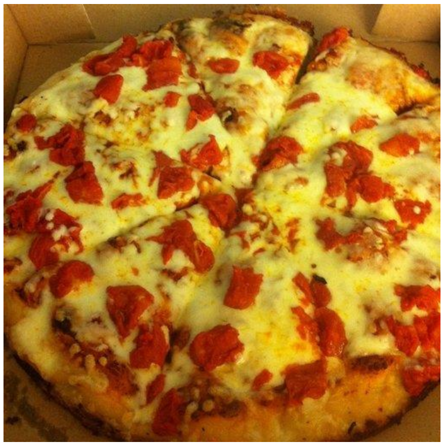
Dataset

For the project, we have downloaded the dataset from Kaggle from the link https://www.kaggle.com/datasets/carlosrunner/pizza-not-pizza/data. The data set has two files namely "Pizza" and "Not Pizza" with 983 images each. All images were rescaled to have a maximum side length of 512 pixels. The training dataset contains 1376 images, Validation dataset contains 294 images, and Testing dataset contains 296 images. To make the model more robust and accurate we performed Data Augmentation to dataset.

In this project we have performed various techniques on the dataset to make the data even and get higher accuracy. The techniques used are resizing, Normalizing, Data Augmenting and again in the augmentation the techniques used are rotation, flipping, zooming and shear transformations. All the images are resized to get a consistent resolution of 224x224 pixels. To create variations of the training data we have applied data augmentation techniques.

Figure 1

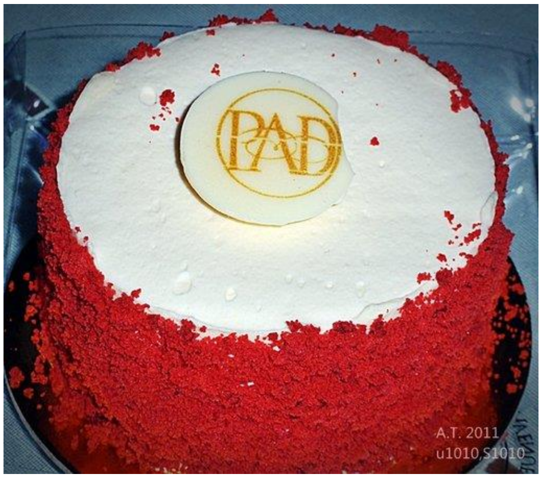
Pizza



Note: This is a picture from the pizza file from the dataset.

Figure 2

Not pizza



Note: This is a picture from the not pizza file from the dataset.

Deliverables

In this project we are comparing two different models namely convolutional neural network and residual Networks. We have chosen these two models to classify the images, and we are trying to select the best model with high accuracy.

As of now we have built a Convolutional Neural Network (CNN) Model, For Binary Classification Of "Pizza" And "Not Pizza, “for the mid-term presentation. We are planning to build a Residual Network Model for our final presentation and then compare the results with the Convolutional Neural Network (CNN) Model. According to the mid-term presentation results we are expecting higher accuracy for the Residual Network Model than the CNN model.

We are done with the midterm presentation and the report and have build the CNN model, for our final report we will work on Residual Network Model

The images are not accurately distributed to the training validation and testing data set. We have faced some issues in setting the directories while writing the code. We are planning to create the directories manually in our dataset folder to cope this issue and get the better result.

Techniques

In this project we are using Convolutional Neural Network (CNN) Model and Residual Networks model. And, for the evaluation metrics we have used accuracy. The Convolutional Neural Network (CNN) is a deep neural network which is mainly used for computer vision. They are designed to process and analyze and visualize the data, CNN model will be effective for tasks such as image recognition, image classification, object detection, and image segmentation.

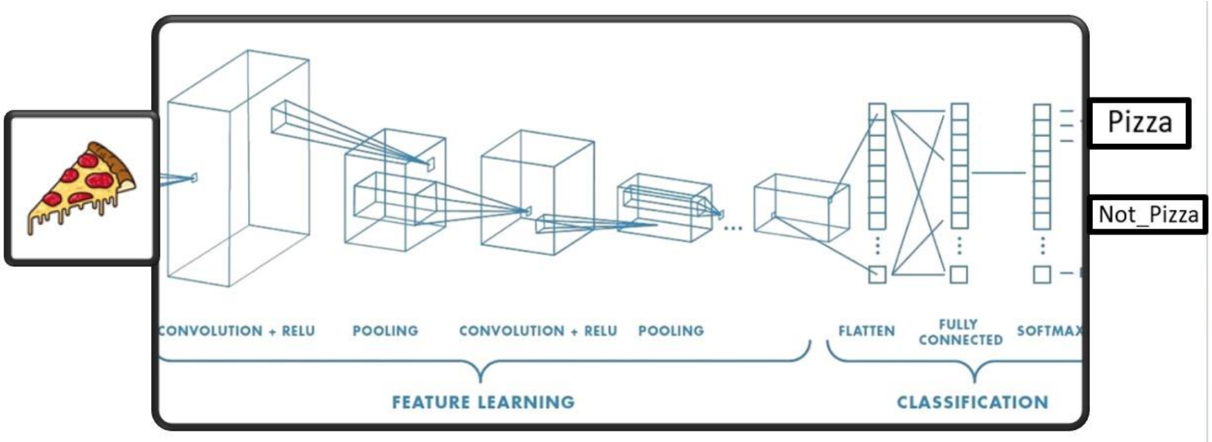
Residual Networks are also called the ResNets, this is another type of deep neural network architecture designed for the gradient problem in very deep convolutional neural networks (CNNs).

According to the code for the CNN model, we got the Training Accuracy as 73.96%, Validation Accuracy as 73.16% and the Test Accuracy as 73.16%.

Design

Figure 3

Convolutional Neural Network – Architecture



Note: The above diagram shows the Conventional Neural Network structure with the layer which we have used in our project

According to the CNN structure from figure 3, we created a sequential model. The ReLU activation function is used for the first Convolutional Layer. The first layer has 32 filters of size 3x3 in it. Added the first Max Pooling Layer with a 2x2 pool size to decrease the spatial dimensions of the feature maps. Added the second Convolutional Layer with 64 filters of size 3x3 and ReLU activation. And then, Add the second Max Pooling Layer with a 2x2 pool size followed by the Fully Connected Layer with 128 neurons and a ReLU activation function. Added the Output Layer with two neurons, "pizza" and "not pizza", and used SoftMax activation for the probability scores.

We used a batch size of 32 in each pass during the training. To fetch the proper accuracy we have divided the dataset into Training, Validation, and Test dataset in the ratio 70 : 15 : 15. We used 10 epochs in the training.

Residual Network Model Building

In this project, as a part of the final presentation, used a Residual Network, or ResNet. ResNet is a type of deep neural network architecture which was introduced to address the problem of existing gradients in the deep neural networks. Usually, the vanishing gradients will occur when they train deep networks and make it challenging for the network to learn productively because the gradients become small when it gets closer to zero.

The key innovation of ResNet is to introduce skip connections or shortcuts. shortcuts generally allow the network to skip the layers during forward and backward propagation. However, the gradient will flow easily through the network, and will become easier to train the deepest networks.

These are import statements for necessary Keras modules, including Model for defining the model, various layers such as Input, Conv2D, BatchNormalization, Activation, Add, GlobalAveragePooling2D, and Dense, and ModelCheckpoint for saving the best model during training.

The provided Python code will defines the binary classification ResNet model using Keras. It consists of a series of residual blocks with convolutional layers, batch normalization, and ReLU activations. The model is built using an initial convolutional layer, followed by three sets of residual blocks with increasing filters and down sampling. Global average pooling is applied, and the model outputs a binary classification using a sigmoid activation. The code assumes the existence of data generators and can be trained using an optimizer, binary crossentropy loss, and potentially saved using the ModelCheckpoint callback.

Results

Based on the results obtained from the two models we have found that the training accuracy is 75.44% and 80.67%; Validation accuracy is 80.61% and 86.39 and Test accuracy is 79.73% and 81.08% respectively for the CNN and RESNET.

Table 3

Evaluation Metrics

EVALUATION METRICS	CNN	RESNET
Training Accuracy	75.44%	80.67%
Validation Accuracy	80.61%	86.39%
Test Accuracy	79.73%	81.08%

Note: The above table is the comparison of two models.

Implementation

For this project we used python language in Jupyter notebook, we have imported libraries like TensorFlow, Keras, Matplotlib, Pandas, NumPy.

Conclusion

In this project, the Resnet model demonstrated superior performance over the traditional CNN model, showcasing its ability to effectively capture intricate features and address the challenges of deep learning tasks, ultimately resulting in enhanced accuracy and predictive capabilities.

References

Du, C.-J., & Sun, D.-W. (2004, July 28). *Comparison of three methods for classification of pizza topping using different colour space transformations*. Journal of Food Engineering. https://www.sciencedirect.com/science/article/abs/pii/S026087740400281X

Du, C.-J., & Sun, D.-W. (2007, October 6). *Multi-classification of pizza using computer vision and Support Vector Machine*. Journal of Food Engineering. https://www.sciencedirect.com/science/article/abs/pii/S0260877407005146

Lu, Y. (2019, February 25). *Food image recognition by using convolutional neural networks (cnns)*. arXiv.org. https://arxiv.org/abs/1612.00983

Dim, P. P. (n.d.). How to make a pizza: Learning a compositional layer-based Gan Model. https://openaccess.thecvf.com/content_CVPR_2019/papers/Papadopoulos_How_to_Make_a_Pizza_Learning_a_Compositional_Layer-Based_GAN_CVPR_2019_paper.pdf?source=post_page---------------------------