**Aim**

to predict diet quality and health outcomes and inform dietary guidelines and product development.

**Objective**

1.Removal of unwanted matter

2.Making food safe for consumption

3.Increasing digestibility

4.Minimizing nutrient loss

5.Increasing acceptability through fabricated food

**ABSTRACT**

The process of identifying food items from an image is quite an interesting field with various applications. Since food monitoring plays a leading role in health-related problems,it is becoming more essential in our day-to-day lives. In this paper, an approach has been presented to classify images of food using convolutional neural networks. Unlike the traditional artiﬁcial neural networks, convolutional neural networks have the capability of estimating the score function directly from image pixels. A 2D convolution layer has been utilized which creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. There are multiple such layers, and the outputs are concatenated at parts to form the ﬁnal tensor of outputs. We also use the Max-Pooling function for the data, and the features extracted from this function are used to train the network. An accuracy of 86.97% for the classes of the FOOD-101 dataset is recognized using the proposed implementation

**INTRODUCTION**

In the current age, people are more conscious about their food and diet to avoid either upcoming or existing diseases. Since people are dependent on smart technologies, provision of an application to automatically monitor the individuals diet,helps in many aspects. It increases the awareness of people in their food habits and diet. Over the last two decades,research has been focused on automatically recognizing the food and their nutritional information from images captured using computer vision and machine learning techniques. In order to properly assess dietary intake, accurate estimation of calorie value of food is of paramount importance. A majority of the people are overeating and not being active enough.Given how busy and stressed people are today, it’s effort less to forget to keep track of the food that they eat. This only increases the importance of proper classiﬁcation of food. As it is frequently said, “we eat with our eyes”. With the continued proliferation of social media platforms such as Instagram (now at 500 million daily active users) as avenues for experience sharing and marketing, our digital experience becomes more and more photo-driven, and of these, over 360 million photos are photos of food. Food images almost single-handedly drive dining experiences, food festivals, cooking classes, and the rise of gastro-tourism, with over 88% of respondents in a 2015 survey considering food to be the defining element in selecting travel destinations. Most of these photos may be associated with a location or a tag, but are otherwise unlabeled, making the food search experience largely disorganized and difficult to navigate. This project explores food image classification with convolutional neural networks (CNNs) for better image labeling and clustering by dish, which in turn may improve the recommendation and search flows for a better digital food user experience overall. Specifically, the goal of the project is to, given an image of a dish as the input to the model, output the correct label categorization of the food image. Food image recognition and calorie estimation can aid in diet management, food blogging and recognizing the Indian foods.

**ARTIFICIAL INTELLIGENCE**

Artificial intelligence (AI), sometimes called machine intelligence, is [intelligence](https://en.wikipedia.org/wiki/Intelligence" \o "Intelligence) demonstrated by [machines](https://en.wikipedia.org/wiki/Machine" \o "Machine), in contrast to the natural intelligence displayed by humans. Leading AI textbooks define the field as the study of "[intelligent agents](https://en.wikipedia.org/wiki/Intelligent_agent" \o "Intelligent agent)": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the [human mind](https://en.wikipedia.org/wiki/Human_mind" \o "Human mind), such as "learning" and "problem solving".

Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans. Some of the activities computers with artificial intelligence are designed for include:

Speech recognition

Learning

Planning

Problem solving

DEFINITIONS

Computer science defines AI research as the study of "[intelligent agents](https://en.wikipedia.org/wiki/Intelligent_agent" \o "Intelligent agent)": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals.[[1]](https://en.wikipedia.org/wiki/Artificial_intelligence" \l "cite_note-Definition_of_AI-1) A more elaborate definition characterizes AI as “a system’s ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation

Artificial intelligence is a branch of computer science that aims to create intelligent machines. It has become an essential part of the technology industry.

Research associated with artificial intelligence is highly technical and specialized. The core problems of artificial intelligence include programming computers for certain traits such as:

Knowledge

Reasoning

Problem solving

Perception

Learning

Planning

**ABILITY TO MANIPULATE AND MOVE OBJECTS**

Knowledge engineering is a core part of AI research. Machines can often act and react like humans only if they have abundant information relating to the world. Artificial intelligence must have access to objects, categories, properties and relations between all of them to implement knowledge engineering. Initiating common sense, reasoning and problem-solving power in machines is a difficult and tedious task.

Machine learning is also a core part of AI. Learning without any kind of supervision requires an ability to identify patterns in streams of inputs, whereas learning with adequate supervision involves classification and numerical regressions. Classification determines the category an object belongs to and regression deals with obtaining a set of numerical input or output examples, thereby discovering functions enabling the generation of suitable outputs from respective inputs. Mathematical analysis of machine learning algorithms and their performance is a well-defined branch of theoretical computer science often referred to as computational learning theory.

Machine perception deals with the capability to use sensory inputs to deduce the different aspects of the world, while computer vision is the power to analyze visual inputs with a few sub-problems such as facial, object and gesture recognition. Robotics is also a major field related to AI. Robots require intelligence to handle tasks such as object manipulation and navigation, along with sub-problems of localization, motion planning and mapping.

**BASICS**

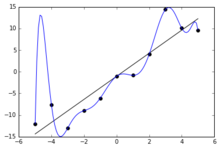
A typical AI analyzes its environment and takes actions that maximize its chance of success. An AI's intended [utility function (or goal)](https://en.wikipedia.org/wiki/Utility_function" \o "Utility function) can be simple ("1 if the AI wins a game of [Go](https://en.wikipedia.org/wiki/Go_(game)" \o "Go (game)), 0 otherwise") or complex ("Do mathematically similar actions to the ones succeeded in the past"). Goals can be explicitly defined, or induced. If the AI is programmed for "[reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning" \o "Reinforcement learning)", goals can be implicitly induced by rewarding some types of behavior or punishing others. Alternatively, an evolutionary system can induce goals by using a "[fitness function](https://en.wikipedia.org/wiki/Fitness_function" \o "Fitness function)" to mutate and preferentially replicate high-scoring AI systems, similarly to how animals evolved to innately desire certain goals such as finding food. Some AI systems, such as nearest-neighbor, instead of reason by analogy, these systems are not generally given goals, except to the degree that goals are implicit in their training data. Such systems can still be benchmarked if the non-goal system is framed as a system whose "goal" is to successfully accomplish its narrow classification task.

AI often revolves around the use of [algorithms](https://en.wikipedia.org/wiki/Algorithms" \o "Algorithms). An algorithm is a set of unambiguous instructions that a mechanical computer can execute. A complex algorithm is often built on top of other, simpler, algorithms. A simple example of an algorithm is the following (optimal for first player) recipe for play at [tic-tac-toe](https://en.wikipedia.org/wiki/Tic-tac-toe" \o "Tic-tac-toe)

Many AI algorithms are capable of learning from data; they can enhance themselves by learning new [heuristics](https://en.wikipedia.org/wiki/Heuristic_(computer_science)" \o "Heuristic (computer science)) (strategies, or "rules of thumb", that have worked well in the past), or can themselves write other algorithms. Some of the "learners" described below, including Bayesian networks, decision trees, and nearest-neighbor, could theoretically, (given infinite data, time, and memory) learn to approximate any [function](https://en.wikipedia.org/wiki/Function_(mathematics)" \o "Function (mathematics)), including which combination of mathematical functions would best describe the world. These learners could therefore, derive all possible knowledge, by considering every possible hypothesis and matching them against the data. In practice, it is almost never possible to consider every possibility, because of the phenomenon of "[combinatorial explosion](https://en.wikipedia.org/wiki/Combinatorial_explosion" \o "Combinatorial explosion)", where the amount of time needed to solve a problem grows exponentially. Much of AI research involves figuring out how to identify and avoid considering broad range of possibilities that are unlikely to be beneficial. For example, when viewing a map and looking for the shortest driving route from [Denver](https://en.wikipedia.org/wiki/Denver" \o "Denver) to [New York](https://en.wikipedia.org/wiki/New_York_City" \o "New York City) in the East, one can in most cases skip looking at any path through [San Francisco](https://en.wikipedia.org/wiki/San_Francisco" \o "San Francisco) or other areas far to the West; thus, an AI wielding a pathfinding algorithm like [A\*](https://en.wikipedia.org/wiki/A*_search_algorithm" \o "A* search algorithm) can avoid the combinatorial explosion that would ensue if every possible route had to be ponderously considered in turn.

The earliest (and easiest to understand) approach to AI was symbolism (such as formal logic): "If an otherwise healthy adult has a fever, then they may have [influenza](https://en.wikipedia.org/wiki/Influenza" \o "Influenza)". A second, more general, approach is [Bayesian inference](https://en.wikipedia.org/wiki/Bayesian_inference" \o "Bayesian inference): "If the current patient has a fever, adjust the probability they have influenza in such-and-such way". The third major approach, extremely popular in routine business AI applications, are analogizers such as [SVM](https://en.wikipedia.org/wiki/Support_vector_machine" \o "Support vector machine) and [nearest-neighbor](https://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm" \o "K-nearest neighbor algorithm): "After examining the records of known past patients whose temperature, symptoms, age, and other factors mostly match the current patient, X% of those patients turned out to have influenza". A fourth approach is harder to intuitively understand, but is inspired by how the brain's machinery works: the [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network" \o "Artificial neural network) approach uses artificial "[neurons](https://en.wikipedia.org/wiki/Neurons" \o "Neurons)" that can learn by comparing itself to the desired output and altering the strengths of the connections between its internal neurons to "reinforce" connections that seemed to be useful. These four main approaches can overlap with each other and with evolutionary systems; for example, neural nets can learn to make inferences, to generalize, and to make analogies. Some systems implicitly or explicitly use multiple of these approaches, alongside many other AI and non-AI algorithms; the best approach is often different depending on the problem.

Learning algorithms work on the basis that strategies, algorithms, and inferences that worked well in the past are likely to continue working well in the future. These inferences can be obvious, such as "since the sun rose every morning for the last 10,000 days, it will probably rise tomorrow morning as well". They can be nuanced, such as "X% of [families](https://en.wikipedia.org/wiki/Family_(biology)" \o "Family (biology)) have geographically separate species with color variants, so there is an Y% chance that undiscovered [black swans](https://en.wikipedia.org/wiki/Black_swan_theory" \o "Black swan theory) exist". Learners also work on the basis of "[Occam's razor](https://en.wikipedia.org/wiki/Occam%27s_razor" \l "Probability_theory_and_statistics" \o "Occam's razor)": The simplest theory that explains the data is the likeliest. Therefore, according to Occam's razor principle, a learner must be designed such that it prefers simpler theories to complex theories, except in cases where the complex theory is proven substantially better.

[](https://en.wikipedia.org/wiki/File:Overfitted_Data.png)

The blue line could be an example of [over fitting](https://en.wikipedia.org/wiki/Overfitting" \o "Overfitting) a linear function due to random noise.

Settling on a bad, overly complex theory gerrymandered to fit all the past training data is known as [over fitting](https://en.wikipedia.org/wiki/Overfitting" \o "Overfitting). Many systems attempt to reduce overfitting by rewarding a theory in accordance with how well it fits the data, but penalizing the theory in accordance with how complex the theory is. Besides classic overfitting, learners can also disappoint by "learning the wrong lesson". A toy example is that an image classifier trained only on pictures of brown horses and black cats might conclude that all brown patches are likely to be horses. A real-world example is that, unlike humans, current image classifiers don't determine the spatial relationship between components of the picture; instead, they learn abstract patterns of pixels that humans are oblivious to, but that linearly correlate with images of certain types of real objects. Faintly superimposing such a pattern on a legitimate image results in an "adversarial" image that the system misclassifies.

[](https://en.wikipedia.org/wiki/File:DÃ©tection_de_personne_-_exemple_3.jpg)

A self-driving car system may use a neural network to determine which parts of the picture seem to match previous training images of pedestrians, and then model those areas as slow-moving but somewhat unpredictable rectangular prisms that must be avoided.

Compared with humans, existing AI lacks several features of human "[commonsense reasoning](https://en.wikipedia.org/wiki/Commonsense_reasoning" \o "Commonsense reasoning)"; most notably, humans have powerful mechanisms for reasoning about "[naïve physics](https://en.wikipedia.org/wiki/Na%C3%AFve_physics" \o "Naïve physics)" such as space, time, and physical interactions. This enables even young children to easily make inferences like "If I roll this pen off a table, it will fall on the floor". Humans also have a powerful mechanism of "[folk psychology](https://en.wikipedia.org/wiki/Folk_psychology" \o "Folk psychology)" that helps them to interpret natural-language sentences such as "The city councilmen refused the demonstrators a permit because they advocated violence". (A generic AI has difficulty discerning whether the ones alleged to be advocating violence are the councilmen or the demonstrators.) This lack of "common knowledge" means that AI often makes different mistakes than humans make, in ways that can seem incomprehensible. For example, existing self-driving cars cannot reason about the location nor the intentions of pedestrians in the exact way that humans do, and instead must use non-human modes of reasoning to avoid accidents.

**CHALLENGES OF AI**

The overall research goal of artificial intelligence is to create technology that allows computers and machines to function in an intelligent manner. The general problem of simulating (or creating) intelligence has been broken down into sub-problems. These consist of particular traits or capabilities that researchers expect an intelligent system to display. The traits described below have received the most attention.

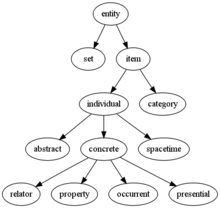
**\**

**REASONING, PROBLEM SOLVING**

Early researchers developed algorithms that imitated step-by-step reasoning that humans use when they solve puzzles or make logical deductions. By the late 1980s and 1990s, AI research had developed methods for dealing with [uncertain](https://en.wikipedia.org/wiki/Uncertainty" \o "Uncertainty) or incomplete information, employing concepts from [probability](https://en.wikipedia.org/wiki/Probability" \o "Probability) and [economics](https://en.wikipedia.org/wiki/Economics" \o "Economics).

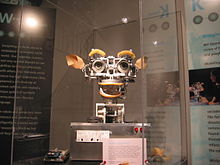
These algorithms proved to be insufficient for solving large reasoning problems, because they experienced a "combinatorial explosion": they became exponentially slower as the problems grew larger. In fact, even humans rarely use the step-by-step deduction that early AI research was able to model. They solve most of their problems using fast, intuitive judgments.

**KNOWLEDGE REPRESENTATION**

[](https://en.wikipedia.org/wiki/File:GFO_taxonomy_tree.png)

An ontology represents knowledge as a set of concepts within a domain and the relationships between those concepts. [Knowledge representation](https://en.wikipedia.org/wiki/Knowledge_representation" \o "Knowledge representation) and knowledge engineering are central to classical AI research. Some "expert systems" attempt to gather together explicit knowledge possessed by experts in some narrow domain. In addition, some projects attempt to gather the "commonsense knowledge" known to the average person into a database containing extensive knowledge about the world. Among the things a comprehensive commonsense knowledge base would contain are: objects, properties, categories and relations between objects; situations, events, states and time; causes and effects; knowledge about knowledge (what we know about what other people know); and many other, less well researched domains. A representation of "what exists" is an [ontology](https://en.wikipedia.org/wiki/Ontology_(computer_science)" \o "Ontology (computer science)): the set of objects, relations, concepts, and properties formally described so that software agents can interpret them. The [semantics](https://en.wikipedia.org/wiki/Semantics" \o "Semantics) of these are captured as [description logic](https://en.wikipedia.org/wiki/Description_logic" \o "Description logic) concepts, roles, and individuals, and typically implemented as classes, properties, and individuals in the [Web Ontology Language](https://en.wikipedia.org/wiki/Web_Ontology_Language" \o "Web Ontology Language). The most general ontologies are called [upper ontologies](https://en.wikipedia.org/wiki/Upper_ontology" \o "Upper ontology), which attempt to provide a foundation for all other knowledge by acting as mediators between [domain ontologies](https://en.wikipedia.org/wiki/Domain_ontology" \o "Domain ontology) that cover specific knowledge about a particular knowledge domain (field of interest or area of concern). Such formal knowledge representations can be used in content-based indexing and retrieval, scene interpretation, clinical decision support, knowledge discovery (mining "interesting" and actionable inferences from large databases), and other areas.

**SOCIAL INTELLIGENCE**

[](https://en.wikipedia.org/wiki/File:Kismet_robot_at_MIT_Museum.jpg)

[Kismet](https://en.wikipedia.org/wiki/Kismet_(robot)" \o "Kismet (robot)), a robot with rudimentary social skills

Moravec's paradox can be extended to many forms of social intelligence. Distributed multi-agent coordination of autonomous vehicles remains a difficult problem. [Affective computing](https://en.wikipedia.org/wiki/Affective_computing" \o "Affective computing) is an interdisciplinary umbrella that comprises systems which recognize, interpret, process, or simulate human [affects](https://en.wikipedia.org/wiki/Affect_(psychology)" \o "Affect (psychology)). Moderate successes related to affective computing include textual [sentiment analysis](https://en.wikipedia.org/wiki/Sentiment_analysis" \o "Sentiment analysis) and, more recently, multimodal affect analysis (see [multimodal sentiment analysis](https://en.wikipedia.org/wiki/Multimodal_sentiment_analysis" \o "Multimodal sentiment analysis)), wherein AI classifies the affects displayed by a videotaped subject.

In the long run, social skills and an understanding of human emotion and [game theory](https://en.wikipedia.org/wiki/Game_theory" \o "Game theory) would be valuable to a social agent. Being able to predict the actions of others by understanding their motives and emotional states would allow an agent to make better decisions. Some computer systems mimic human emotion and expressions to appear more sensitive to the emotional dynamics of human interaction, or to otherwise facilitate [human–computer interaction](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction" \o "Human–computer interaction). Similarly, some [virtual assistants](https://en.wikipedia.org/wiki/Virtual_assistant" \o "Virtual assistant) are programmed to speak conversationally or even to banter humorously; this tends to give naïve users an unrealistic conception of how intelligent existing computer agents actually are.

**GENERAL INTELLIGENCE**

Historically, projects such as the Cyc knowledge base (1984–) and the massive Japanese [Fifth Generation Computer Systems](https://en.wikipedia.org/wiki/Fifth_generation_computer" \o "Fifth generation computer) initiative (1982–1992) attempted to cover the breadth of human cognition. These early projects failed to escape the limitations of non-quantitative symbolic logic models and, in retrospect, greatly underestimated the difficulty of cross-domain AI. Nowadays, the vast majority of current AI researchers work instead on tractable "narrow AI" applications (such as medical diagnosis or automobile navigation). Many researchers predict that such "narrow AI" work in different individual domains will eventually be incorporated into a machine with [artificial general intelligence](https://en.wikipedia.org/wiki/Artificial_general_intelligence" \o "Artificial general intelligence) (AGI), combining most of the narrow skills mentioned in this article and at some point even exceeding human ability in most or all these areas. Many advances have general, cross-domain significance. One high-profile example is that [DeepMind](https://en.wikipedia.org/wiki/DeepMind" \o "DeepMind) in the 2010s developed a "generalized artificial intelligence" that could learn many diverse [Atari](https://en.wikipedia.org/wiki/Atari_2600" \o "Atari 2600) games on its own, and later developed a variant of the system which succeeds at [sequential learning](https://en.wikipedia.org/wiki/Catastrophic_interference" \l "The_Sequential_Learning_Problem:_McCloskey_and_Cohen_(1989)" \o "Catastrophic interference). Besides [transfer learning](https://en.wikipedia.org/wiki/Transfer_learning" \o "Transfer learning), hypothetical AGI breakthroughs could include the development of reflective architectures that can engage in decision-theoretic metareasoning, and figuring out how to "slurp up" a comprehensive knowledge base from the entire unstructured [Web](https://en.wikipedia.org/wiki/World_Wide_Web" \o "World Wide Web). Some argue that some kind of (currently-undiscovered) conceptually straightforward, but mathematically difficult, "Master Algorithm" could lead to AGI. Finally, a few "emergent" approaches look to simulating human intelligence extremely closely, and believe that [anthropomorphic](https://en.wikipedia.org/wiki/Anthropomorphism" \o "Anthropomorphism) features like an [artificial brain](https://en.wikipedia.org/wiki/Artificial_brain" \o "Artificial brain) or simulated [child development](https://en.wikipedia.org/wiki/Developmental_robotics" \o "Developmental robotics) may someday reach a critical point where general intelligence emerges.

Many of the problems in this article may also require general intelligence, if machines are to solve the problems as well as people do. For example, even specific straightforward tasks, like [machine translation](https://en.wikipedia.org/wiki/Machine_translation" \o "Machine translation), require that a machine read and write in both languages ([NLP](https://en.wikipedia.org/wiki/Artificial_intelligence" \l "Natural_language_processing)), follow the author's argument ([reason](https://en.wikipedia.org/wiki/Artificial_intelligence" \l "Deduction,_reasoning,_problem_solving)), know what is being talked about ([knowledge](https://en.wikipedia.org/wiki/Artificial_intelligence" \l "Knowledge_representation)), and faithfully reproduce the author's original intent ([social intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence" \l "Social_intelligence)). A problem like machine translation is considered "[AI-complete](https://en.wikipedia.org/wiki/AI-complete" \o "AI-complete)", because all of these problems need to be solved simultaneously in order to reach human-level machine performance.

**AIM**

Classify the food using artificial intelligence a Deep learning method called “Convolution neural network “ with best accuracy.

**OBJECTIVE**

The main goal of the project is to develop advanced concepts of early Food Classification . Here we provide an automatic prediction of road potholes using advanced Artificial intelligence technology.

**PROBLEM STATEMENT**

According to the survey “Road Accidents in India”, by the ministry of road transport and highways, a total of 1, 42,485 people have lost their lives due to fatal road accidents. Nearly 1.5 percent fatalities were due to poor road conditions. Thus for these fatal problems, a cost effective solution is needed that collects the information about the severity of potholes and humps which helps drivers to drive

safely. With the proposed system an attempt has been made to provide approve drivers to ward off the accidents caused due to potholes .

**METHODOLOGY**

The flowchart of the food classification system using the Convolutional Neural Network is shown in Fig. 1. The food classification system consists of two phases. The first phase is the training phase, in which feature extraction and classification of food images are performed to develop a

prediction model. In the testing phase, this prediction model is used to classify the food test images. Here, CNN performs automatic feature extraction and classification of food images. CNN’s have an input layer, an output layer, and hidden layers, all of which aid in image classification and processing. The convolutional layer is the foundation of CNN. By sliding a kernel over the input images, this layer performs convolution and generates a feature map. The dimension of the feature map is minimized by the pooling layer. Max pooling is the most common type of

pooling. Pooling is achieved by sliding a window over the input and taking the highest value in the window. The output of the last pooling layer is converted to a one-dimensional vector by the FC layer.

**LITERATURE SURVEY**

# **Automated Food image Classification using Deep Learning approach.**

**Sapna Yadav; Alpana; Satish Chand (IEEE-2021)**

Food image classification is an emerging research field due to its increasing benefits in the health and medical sectors. For sure, in the future automated food recognition tools will help in developing diet monitoring systems, calories estimation and so on. In this paper, automated methods of food classification using deep learning approaches are presented. Squeeze Net and VGG-16 Convolution Neural Networks are used for food image classification. It is demonstrated that using data augmentation and by fine-tuning the hyper parameters, these networks exhibited much better performance, making these networks suitable for practical applications in health and medical fields. Squeeze Net being a lightweight network, is easier to deploy and often more desirable. Even with fewer parameters, SqueezeNet is able to achieve quite a good accuracy of 77.20%. Higher accuracy of food image classification is further achieved by extracting complex features of food images. The performance of automatic food image classification is further improved by the proposed VGG-16 network. Due to increased network depth, proposed VGG-16 has achieved significant improvement in accuracy up to 85.07%.

1. **Food Recognition based on Deep Learning Algorithms.**

**Anis Nasuha Mohd Zulfikri(IEEE-2022)**

Accurate methods can help the technology nowadays to keep improving and provide a reliable system for the people to use. In this paper, two different image classification systems; Convolutional Neural Network (CNN) and Residual Neural Network (ResNet) were proposed in order to recognize six food classes; Apple, Orange, Avocado, Milo, Vico and Koko based on color features. Then, the overall performance for both classifications were analyzed in the end of this paper. Datasets of food images were collected from various sources consisting 400 images for each food classes to test the robustness of each classification system. The data were split into 60% training data, 20% validation data and 20% testing data. The system that is proposed in this paper consist of 4 layers for Convolutional Neural Network (CNN) while Residual Neural Network (ResNet) consist of 50 layers. The color feature extraction that is involved for both classifications, RGB values (Red, Green, Blue) are highly considered in order to determine the category of the food. Overall, this experimental results on food recognition showed 100% training accuracy and 98.67% overall testing accuracy for CNN while 99.87% training accuracy and 96.67% overall testing accuracy for ResNet.

1. **Nutrient Food Prediction Through Deep Learning.**

**Saikat Banerjee(IEEE-2021)**

The lifespan of a man can be sustained only with adequate nourishment. To lead a productive, healthy life, human needs nutritious food. In this pandemic COVID-19 situation humans need more nutritious food for combating infectious disease along with a strong immune system in our body. Nutritious foods recognition is one of the major tasks for a customer. In large stores plenty of agricultural products are stored, then there needs a classification for separating normal food and nutritious food. The real time decision will alert the consumer by predicting nutritious foods. By the use of deep learning, it may be possible to classify nutritious food along with their nutrient content and give the possible particular rating view image through the deep learning method. Enormous development in deep learning is possible due to the advancement of the Convolutional Neural Network (CNN) algorithms. CNN is a modern technique inspired by biological neurons mainly used for image processing and data analysis, producing encouraging results. The principal objective of our work is to detect and segregate normal food and nutritious food. This is accomplished using the combination of both nutrition and image Classification techniques. Hence, the proposed system achieved average overall accuracy is more than 91%.

1. **Food Classification and Recommendation for Health and Diet Tracking using Machine Learning.Dr.Manjula.**

Many researchers have been published recently on food classification and recommendation separately, but combination of food classification and recommendation using deep learning is rare. The CNN algorithm is presented in this work because it is higher accuracy than other algorithms. In the present generation people are very concerned about their food habits in order to maintain their healthy balanced diet. This paper classifies Indian food images. The model/system uses a deep leaning process to train the machine. For this project the dataset is collected from Kaggle, UCI and some of the images from Google chrome, which contains 1000 images. The dataset is classified into 12 classes namely biryani, bisibelebath, butter naan, chats, chapatti, Dhokla, dosa, idly, noodles, upma, poori, samosa. On a different set of tests, the average accuracy is 86.33 percent. This paper also contributes to diabetic patients and also recommends the healthy note.

**EXISTING SYSTEM**

In the Existing system there are limits in assessing healthy and abnormal swallowing by Video fluoroscopic swallowing study. Classification of accelero metric swallowing signals is much more efficient method to judge healthy swallowing. However, these methods have developed mostly with dual axis accelerometer signals and classifying two-class problems. This study is to examine classification methods with multi-class three-axis accelero metric signals. Swallowing signals of five foods are classified with both supervised learning algorithm and unsupervised learning algorithm. Three-axis signals noised by 10-level discrete wavelet transform with soft threshold before feature calculation. The result confirmed that classification with support vector machine and K-nearest neighbor can predict with 90% accuracy. However, Classification with fuzzy c-mean clustering produce low purity and normalized mutual information.

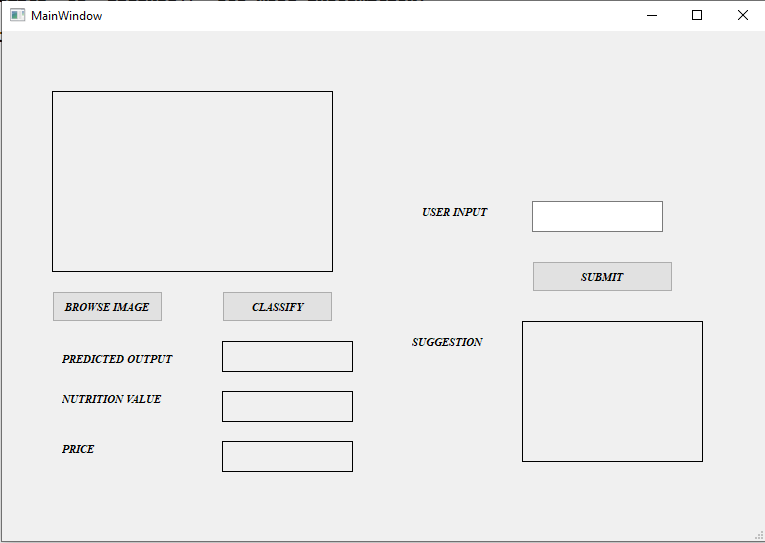
**PROPOSED SYSTEM**

In the proposed system we use new deep learning algorithm called CNN (convolutional nueral network).The process of identifying food items from images is a fascinating field with numerous applications. Food monitoring is becoming more important in our daily lives as it plays a leading role in health-related problems. A method for classifying food images using convolutional neural networks is presented in this paper. Convolutional neural networks, unlike traditional artificial neural networks, can estimate the score function directly from image pixels. A 2D convolution layer was used to generate a convolution kernel, which was convolved with the layer input to generate a tensor of outputs. There are several of these layers, and the outputs are concatenated at different points to form the final tensor of outputs.

**GUI (GRAPHICAL USER INTERFACE)**

**PyQT5**

In this project we use PyQt5 which is used to create a GUI for our food classification project.PyQt is a Python binding for [Qt](https://wiki.qt.io/About_Qt), which is a set of [C++](https://realpython.com/python-vs-cpp/) libraries and development tools providing platform-independent abstractions for graphical user interfaces (GUIs). Qt also provides tools for networking, [threads](https://realpython.com/python-pyqt-qthread/), [regular expressions](https://realpython.com/regex-python/), [SQL databases](https://realpython.com/python-pyqt-database/), [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics), [OpenGL](https://en.wikipedia.org/wiki/OpenGL), [XML](https://en.wikipedia.org/wiki/XML), and many other powerful features.



**PROPOSED BLOCK DIAGRAM**

Dataset, Train, Val

Dataset Pre-processing

Conv2D,Maxpool, Flatten, Dense

Faster R-CNN Model

Faster R-CNN Model

Predicted Output

Testing with a new input image

**BLOCK DIAGRAM EXPLANANTION**

**DATASET**

A huge data set consisting of retina images with very high resolution has been taken with various imaging conditions from Kaggle. 

Training Images

**PREPROCESSING**

The data set was taken from an online platform named Kaggle. The size of the data set was trimmed to 350 images. Before feeding the input directly into the model, the data which is the set of fundus images must undergo some preprocessing steps which includes i. resizing of images size from 3888 \*2951.to 786 \* 786 dimension. ii. perform flip-flop operations which are rotating the fundus image by 90 degrees. Flipping of images is done in order to exercise the model in an efficient way. The input data set is classified into three different categories. They are a. Training dataset, the dataset that is used to train or exercise the model. This data is labeled data set. b. Testing data set, which is used to test the model. c. Validation data set, the dataset that is used to validate the model. Validation data set is used to ensure that the model is not over-fitting whereas training data helps to minimize the loss function. Updating of weights happens accordingly when the training data set is exercised in the model but validation data set does not involve any updating process. Training dataset and validation dataset are labeled but not the testing dataset. Also, one hot encoding is performed on the training labels.

**Training and Testing Data**

After preprocessing is done, the dataset is divided into two parts as Training and Testing.

The training data is used to train the model whereas, the testing data is used to validate the model.

**MODEL ARCHITECTURE - CNN Layers**

The architecture consists of six layers: 1 Input layer, 2 pairs of Convolutional layers & Max Pooling Layers, 1 Output Layer.

1. Input Layer This layer consists of 786 x 786 neurons which is equal to the count of pixels of each individual image being passed. Here, the pixels values of the training images are sent to the input layer.

2. Convolutional Layer 1 This layer consists of 32 neurons. There is a connection between each of the neurons present in this layer to all of the neurons in the previous layer. Convolution is performed on the input pixels, which is a process of performing dot product on the pixel values with arbitrary numbers called as filters. So, the layer’s output is further passed to the max pooling layer.

3. Max Pooling Layer 1 With the filters provided max pooling operation is performed on the received input which is identification of highest value in each patch of feature map.

4. Convolution Layer 2 The max pooling layer’s output is concatenated to a convolution layer (convolution layer 2) with 16 filters, kernel size as 4\*4 and activation function as ReLu. This layer is further passed to a max pooling layer.

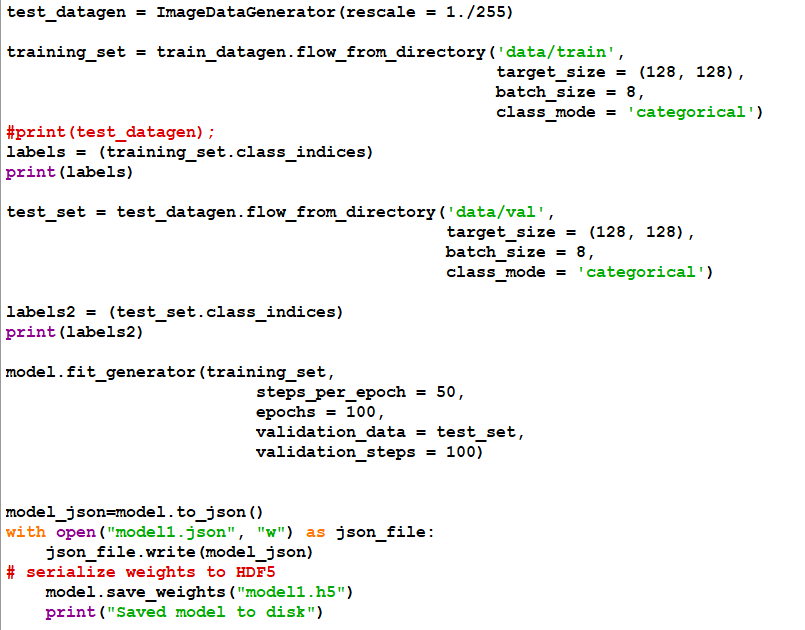
5. Max Pooling Layer 2 The max pooling layer performs the max pooling operation on the received input. Then the output of the max pooling layer is flattened. Flattening is a process of converting any matrix into one dimensional array. Flatten function is applied on the convolution layer to create a single long feature vector.

6. Output Layer The total amount of neurons existing in this layer is equal to the number of levels the disease is classified into. The neuron consisting of the maximum value ranging between 0-1 will be the output i.e., the level in which the disease is. This output will be compared with the actual values and the error is determined. Based on the error the model tunes its underline parameters such that the error is as minimum as possible. This operation is performed on each and every training image.

**CODE**

**TRAINING CODE**





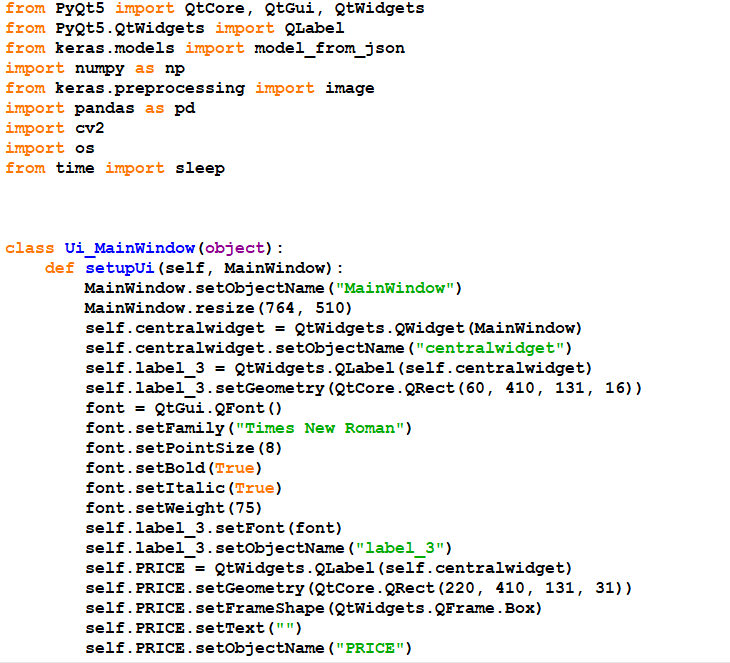
**PREDICTION**

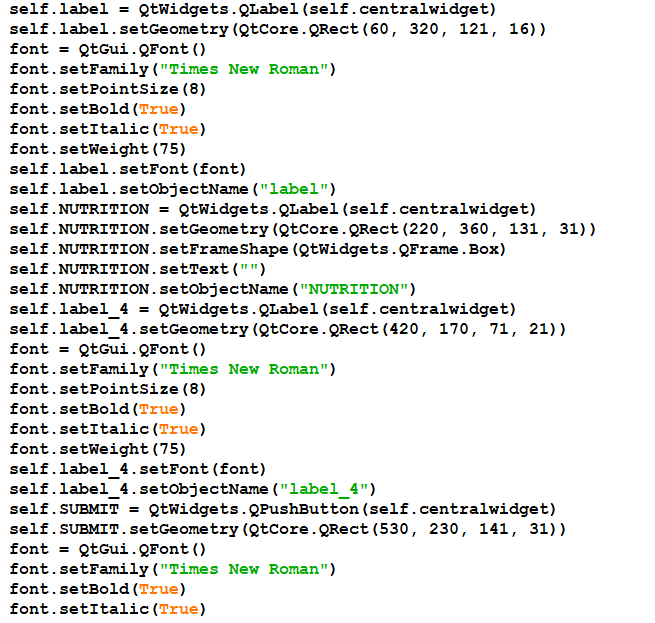
The built model is now evaluated using the testing data and the accuracy is computed which acts as the performance metric of the model.

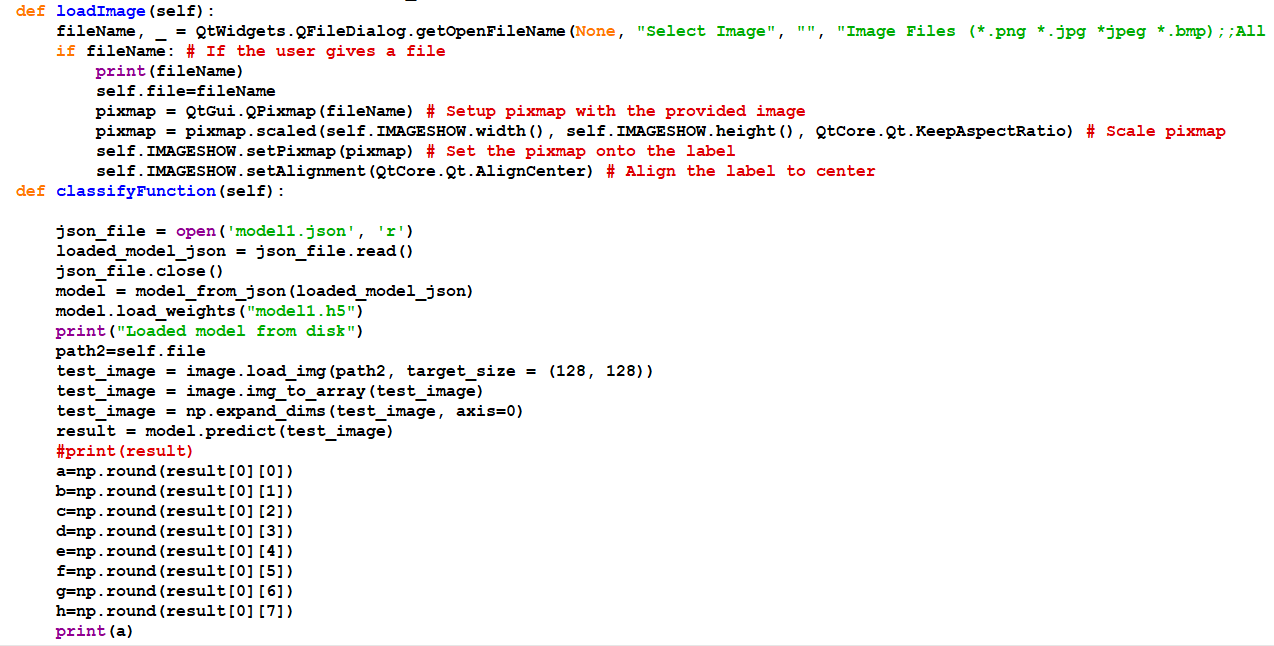
**TESTING MODEL**

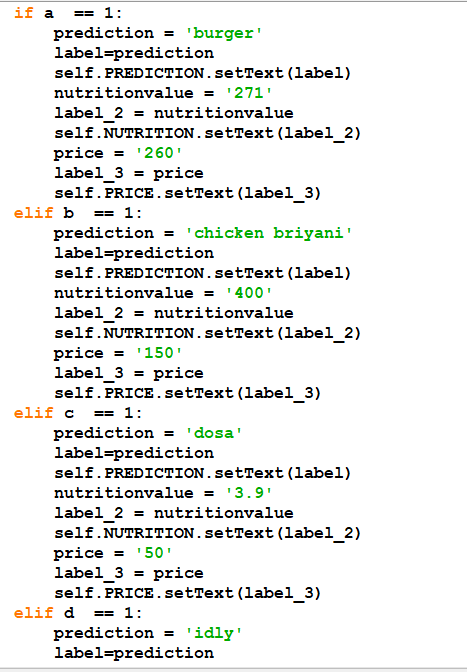
I tested my CNN model on 1622 images. I had an validation accuracy of 97.22 %. My model has a precision of 87.31 % and recall of 74.46 %. The model has a specificity of 97.68 %.

From the confusion matrix, we can conclude that out of 1622 snow, fog and vehicle images of are detected.









**SOFTWARE REQUIREMETS**

**HARDWARE SPECIFICATION**

|  |  |  |
| --- | --- | --- |
| System | : | PC OR LAPTOP |
| Processor | : | INTEL I5 |
| RAM | : | 4 GB Recommended |
| ROM | : | 2 GB |
| FRONT END | : | PYTHON SHELL |
| BACKEND | : | PYTHON SCRIPT WINDOW |
| LAPTOP | : | CAMERA |

**SOFTWARE SPECIFICATION**

|  |  |  |
| --- | --- | --- |
| OPERATING SYSTEM | : | WINDOWS 7/10/11 |
| LANGUAGE USED | : | PYTHON |

**CLASS DIAGRAM:**

**DATASET**

**TRAINING**

**TESTING**

**CNN ALGORITHM**

**TENSOR FLOW RECORD**

**CLASSIFICATION**

**NEW DATA**

**PREDICTED OUTCOME**

**TENSOR FLOW RECORD**

**SEQUENCE DIAGRAM**

**PREDICTING OUTCOME**

**CNN ALGORITHM**

**image**

**TENSOR FLOW RECORD**

****

****

****

**Camera processing classifying each Frame CNN Working**

**OUTPUT**

**SOFTWARE DISCRIPTION**

* **PYTHON**
* **KERAS**
* **TENSORFLOW**
* **NUMPY**
* **PILLOW**
* **CONVOLUTIONAL NEURAL NETWORK**

**PYTHON**

Python is a wonderful and powerful programming language that's easy to use (easy to read **and** write) and with Raspberry Pi lets you connect your project to the real world.

Python syntax is very clean, with an emphasis on readability and uses standard English keywords. Start by opening IDLE from the desktop.

**IDLE**

The easiest introduction to Python is through IDLE, a Python development environment. Open IDLE from the Desktop or applications menu:

IDLE gives you a REPL (Read-Evaluate-Print-Loop) which is a prompt you can enter Python commands in to. As it's a REPL you even get the output of commands printed to the screen without using print.

>>> 1 + 2

3

>>>name = "Sarah"

>>> "Hello " + name

'Hello Sarah'

IDLE also has syntax highlighting built in and some support for autocompletion. You can look back on the history of the commands you've entered in the REPL with Alt + P (previous) and Alt + N (next).

**BASIC PYTHON USAGE**

Hello world in Python:

print("Hello world")

Simple as that!

**INDENTATION**

Some languages use curly braces { and } to wrap around lines of code which belong together, and leave it to the writer to indent these lines to appear visually nested. However, Python does not use curly braces but instead requires indentation for nesting. For example a for loop in Python:

fori in range(10):

print("Hello")

The indentation is necessary here. A second line indented would be a part of the loop, and a second line not indented would be outside of the loop. For example:

fori in range(2):

print("A")

print("B")

would print:

A

B

A

B

whereas the following:

fori in range(2):

print("A")

print("B")

would print:

A

A

B

**VARIABLES**

To save a value to a variable, assign it like so:

name = "Bob"

age = 15

Note here I did not assign types to these variables, as types are inferred, and can be changed (it's dynamic).

age = 15

age += 1 # increment age by 1

print(age)

This time I used comments beside the increment command.

**COMMENTS**

Comments are ignored in the program but there for you to leave notes, and are denoted by the hash # symbol. Multi-line comments use triple quotes like so:

"""

This is a very simple Python program that prints "Hello".

That's all it does.

"""

print("Hello")

**LISTS**

Python also has lists (called arrays in some languages) which are collections of data of any type:

numbers = [1, 2, 3]

Lists are denoted by the use of square brackets [] and each item is separated by a comma.

**ITERATION**

Some data types are iterable, which means you can loop over the values they contain. For example a list:

numbers = [1, 2, 3]

for number in numbers:

print(number)

This takes each item in the list numbers and prints out the item:

1

2

3

Note I used the word number to denote each item. This is merely the word I chose for this - it's recommended you choose descriptive words for variables - using plurals for lists, and singular for each item makes sense. It makes it easier to understand when reading.

Other data types are iterable, for example the string:

dog\_name = "BINGO"

for char in dog\_name:

print(char)

This loops over each character and prints them out:

B

I

N

G

O

**RANGE**

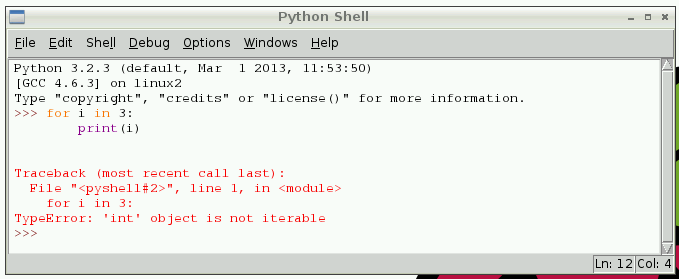
The integer data type is not iterable and trying to iterate over it will produce an error. For example:

fori in 3:

print(i)

will produce:

TypeError: 'int' object is not iterable



However you can make an iterable object using the range function:

fori in range(3):

print(i)

range(5) contains the numbers 0, 1, 2, 3 and 4 (five numbers in total). To get the numbers 1 to 5 use range(1, 6).

**LENGTH**

You can use functions like len to find the length of a string or a list:

name = "Jamie"

print(len(name)) # 5

names = ["Bob", "Jane", "James", "Alice"]

print(len(names)) # 4

**IF STATEMENTS**

You can use if statements for control flow:

name = "Joe"

iflen(name) > 3:

print("Nice name,")

print(name)

else:

print("That's a short name,")

print(name)

**PYTHON FILES IN IDLE**

To create a Python file in IDLE, click File > New File and you'll be given a blank window. This is an empty file, not a Python prompt. You write a Python file in this window, save it, then run it and you'll see the output in the other window.

For example, in the new window, type:

n = 0

fori in range(1, 101):

n += i

print("The sum of the numbers 1 to 100 is:")

print(n)

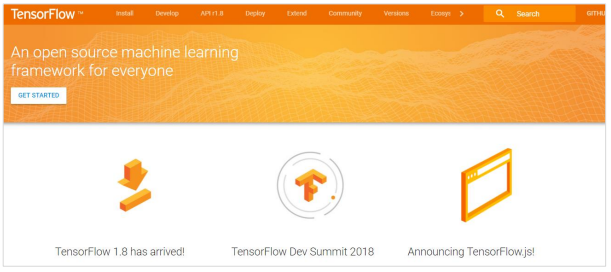
Then save this file (File > Save or Ctrl + S) and run (Run > Run Moduleor hit F5) and you'll see the output in your original Python window.

**EXECUTING PYTHON FILES FROM THE COMMAND LINE**

You can write a Python file in a standard [editor](https://www.raspberrypi.org/documentation/linux/usage/text-editors.md) like Vim, Nano or LeafPad, and run it as a Python script from the command line. Just navigate to the directory the file is saved (use cd and ls for guidance) and run with python, e.g. python hello.py.

**TENSORFLOW-** **INTRODUCTION**

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions. The official website of TensorFlow is mentioned below: <https://www.tensorflow.org/>



Let us now consider the following important features of TensorFlow:

• It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.

• It includes a programming support of deep neural networks and machine learning techniques.

• It includes a high scalable feature of computation with various data sets.

• TensorFlow uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used.

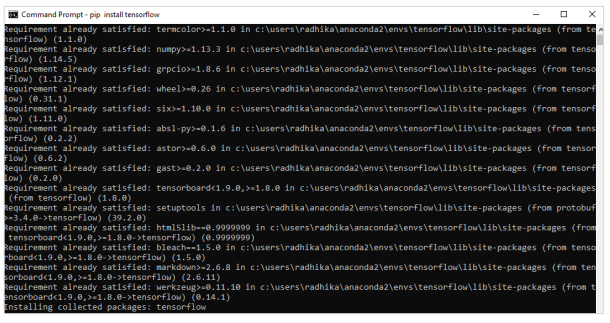
**Why is TensorFlow So Popular?**

TensorFlow is well-documented and includes plenty of machine learning libraries. It offers a few important functionalities and methods for the same. TensorFlow is also called a “Google” product. It includes a variety of machine learning and deep learning algorithms. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding and creation of various sequence models.

**TensorFlow — Installation**

To install TensorFlow, it is important to have “Python” installed in your system. Python version 3.4+ is considered the best to start with TensorFlow installation. Consider the following steps to install TensorFlow in Windows operating system.

**pip install tensorflow**

****

**TensorFlow — Convolutional Neural Networks**

After understanding machine-learning concepts, we can now shift our focus to deep learning concepts. Deep learning is a division of machine learning and is considered as a crucial step taken by researchers in recent decades. The examples of deep learning implementation include applications like image recognition and speech recognition.

Following are the two important types of deep neural networks:

• Convolutional Neural Networks

• Recurrent Neural Networks In this chapter, we will focus on the CNN, Convolutional Neural Networks

**Convolutional Neural Networks**

Convolutional Neural networks are designed to process data through multiple layers of arrays. This type of neural networks is used in applications like image recognition or face recognition. The primary difference between CNN and any other ordinary neural network is that CNN takes input as a two-dimensional array and operates directly on the images rather than focusing on feature extraction which other neural networks focus on. The dominant approach of CNN includes solutions for problems of recognition. Top companies like Google and Facebook have invested in research and development towards recognition projects to get activities done with greater speed.

A convolutional neural network uses three basic ideas:

• Local respective fields

• Convolution

• Pooling

Let us understand these ideas in detail.

CNN utilizes spatial correlations that exist within the input data. Each concurrent layer of a neural network connects some input neurons. This specific region is called local receptive field. Local receptive field focusses on the hidden neurons. The hidden neurons process the input data inside the mentioned field not realizing the changes outside the specific boundary.

If we observe the above representation, each connection learns a weight of the hidden neuron with an associated connection with movement from one layer to another. Here, individual neurons perform a shift from time to time. This process is called “convolution”. The mapping of connections from the input layer to the hidden feature map is defined as “shared weights” and bias included is called “shared bias”. CNN or convolutional neural networks use pooling layers, which are the layers, positioned immediately after CNN declaration. It takes the input from the user as a feature map that comes out of convolutional networks and prepares a condensed feature map. Pooling layers helps in creating layers with neurons of previous layers.

**KERAS**

**INTRODUCTION**

Deep learning is one of the major subfield of machine learning framework. Machine learning is the study of design of algorithms, inspired from the model of human brain. Deep learning is becoming more popular in data science fields like robotics, artificial intelligence(AI), audio & video recognition and image recognition. Artificial neural network is the core of deep learning methodologies. Deep learning is supported by various libraries such as Theano, TensorFlow, Caffe, Mxnet etc., Keras is one of the most powerful and easy to use python library, which is built on top of popular deep learning libraries like TensorFlow, Theano, etc., for creating deep learning models.

**OVERVIEW OF KERAS**

Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

**FEATURES**

Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features:

• Consistent, simple and extensible API.

• Minimal structure - easy to achieve the result without any frills.

• It supports multiple platforms and back ends.

• It is user friendly framework which runs on both CPU and GPU.

• Highly scalability of computation.

**BENEFITS**

Keras is highly powerful and dynamic framework and comes up with the following advantages:

• Larger community support.

• Easy to test.

• Keras neural networks are written in Python which makes things simpler.

• Keras supports both convolution and recurrent networks.

• Deep learning models are discrete components, so that, you can combine into many ways.

**KERAS ― OVERVIEW OF DEEP LEARNING**

Deep learning is an evolving sub field of machine learning. Deep learning involves analyzing the input in layer by layer manner, where each layer progressively extracts higher level information about the input. Let us take a simple scenario of analyzing an image. Let us assume that your input image is divided up into a rectangular grid of pixels. Now, the first layer abstracts the pixels. The second layer understands the edges in the image. The Next layer constructs nodes from the edges. Then, the next would find branches from the nodes. Finally, the output layer will detect the full object. Here, the feature extraction process goes from the output of one layer into the input of the next subsequent layer. By using this approach, we can process huge amount of features, which makes deep learning a very powerful tool. Deep learning algorithms are also useful for the analysis of unstructured data. Let us go through the basics of deep learning in this chapter.

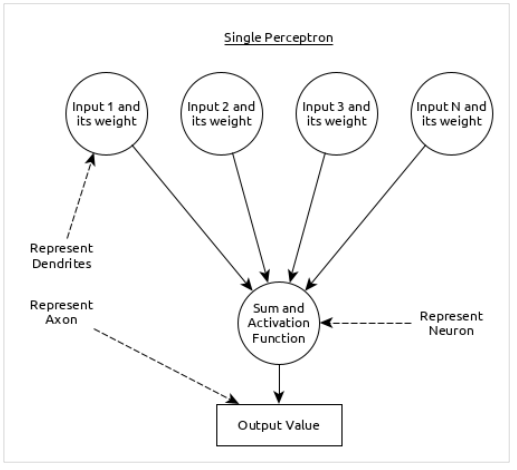
**Artificial Neural Networks**

The most popular and primary approach of deep learning is using “Artificial neural network” (ANN). They are inspired from the model of human brain, which is the most complex organ of our body. The human brain is made up of more than 90 billion tiny cells called “Neurons”. Neurons are inter-connected through nerve fiber called “axons” and “Dendrites”. The main role of axon is to transmit information from one neuron to another to which it is connected.

Similarly, the main role of dendrites is to receive the information being transmitted by the axons of another neuron to which it is connected. Each neuron processes a small information and then passes the result to another neuron and this process continues. This is the basic method used by our human brain to process huge about of information like speech, visual, etc., and extract useful information from it.

Based on this model, the first Artificial Neural Network (ANN) was invented by psychologist Frank Rosenblatt, in the year of 1958. ANNs are made up of multiple nodes which is similar to neurons. Nodes are tightly interconnected and organized into different hidden layers. The input layer receives the input data and the data goes through one or more hidden layers sequentially and finally the output layer predict something useful about the input data. For example, the input may be an image and the output may be the thing identified in the image, say a “Cat”.

A single neuron (called as perceptron in ANN) can be represented as below:

****

Here,

• Multiple input along with weight represents dendrites.

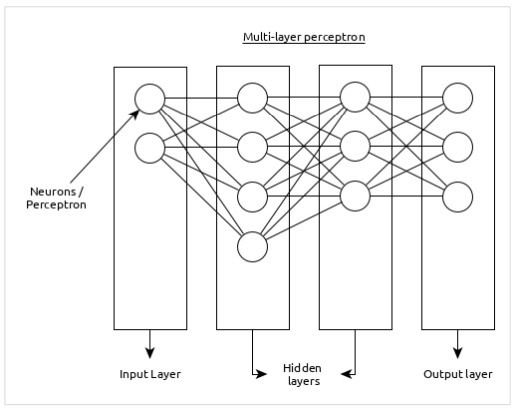
• Sum of input along with activation function represents neurons. Sum actually means computed value of all inputs and activation function represent a function, which modify the Sum value into 0, 1 or 0 to 1.

• Actual output represent axon and the output will be received by neuron in next layer.

Let us understand different types of artificial neural networks in this section.

**Multi-Layer Perceptron**

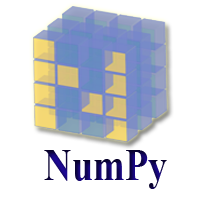
Multi-Layer perceptron is the simplest form of ANN. It consists of a single input layer, one or more hidden layer and finally an output layer. A layer consists of a collection of perceptron. Input layer is basically one or more features of the input data. Every hidden layer consists of one or more neurons and process certain aspect of the feature and send the processed information into the next hidden layer. The output layer process receives the data from last hidden layer and finally output the result.

****

**PYTHON NUMPY**

Our Python NumPy Tutorial provides the basic and advanced concepts of the NumPy. Our NumPy tutorial is designed for beginners and professionals.

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements.



**What is NumPy**

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements.

Travis Oliphant created NumPy package in 2005 by injecting the features of the ancestor module Numeric into another module Numarray. It is an extension module of Python which is mostly written in C. It provides various functions which are capable of performing the numeric computations with a high speed.

NumPy provides various powerful data structures, implementing multi-dimensional arrays and matrices. These data structures are used for the optimal computations regarding arrays and matrices. In this tutorial, we will go through the numeric python library NumPy.

**The need of NumPy**

With the revolution of data science, data analysis libraries like NumPy, SciPy, Pandas, etc. have seen a lot of growth. With a much easier syntax than other programming languages, python is the first choice language for the data scientist.

NumPy provides a convenient and efficient way to handle the vast amount of data. NumPy is also very convenient with Matrix multiplication and data reshaping. NumPy is fast which makes it reasonable to work with a large set of data.

There are the following advantages of using NumPy for data analysis.

1. NumPy performs array-oriented computing.
2. It efficiently implements the multidimensional arrays.
3. It performs scientific computations.
4. It is capable of performing Fourier Transform and reshaping the data stored in multidimensional arrays.
5. NumPy provides the in-built functions for linear algebra and random number generation.

Nowadays, NumPy in combination with SciPy and Mat-plotlib is used as the replacement to MATLAB as Python is more complete and easier programming language than MATLAB.

**Prerequisite**

Before learning Python Numpy, you must have the basic knowledge of Python concepts.

**PYTHON PILLOW ― OVERVIEW**

In today’s digital world, we come across lots of digital images. In case, we are working with Python programming language, it provides lot of image processing libraries to add image processing capabilities to digital images.

Some of the most common image processing libraries are: OpenCV, Python Imaging Library (PIL), Scikit-image, Pillow. However, in this tutorial, we are only focusing on Pillow module and will try to explore various capabilities of this module.

Pillow is built on top of PIL (Python Image Library). PIL is one of the important modules for image processing in Python. However, the PIL module is not supported since 2011 and doesn’t support python 3.

Pillow module gives more functionalities, runs on all major operating system and support for python 3. It supports wide variety of images such as “jpeg”, “png”, “bmp”, “gif”, “ppm”, “tiff”. You can do almost anything on digital images using pillow module. Apart from basic image processing functionality, including point operations, filtering images using built-in convolution kernels, and color space conversions.

**IMAGE ARCHIVES**

The Python Imaging Library is best suited for image archival and batch processing applications. Python pillow package can be used for creating thumbnails, converting from one format to another and print images, etc.

**IMAGE DISPLAY**

You can display images using Tk PhotoImage, BitmapImage and Windows DIB interface, which can be used with PythonWin and other Windows-based toolkits and many other Graphical User Interface (GUI) toolkits.

For debugging purposes, there is a show () method to save the image to disk which calls the external display utility.

**Image Processing**

The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation.

Pillow module allows you to pull some statistics data out of image using histogram method, which later can be used for statistical analysis and automatic contrast enhancement.

**PYTHON PILLOW — ENVIRONMENT SETUP**

This chapter discusses how to install pillow package in your computer.

Installing pillow package is very easy, especially if you’re installing it using pip.

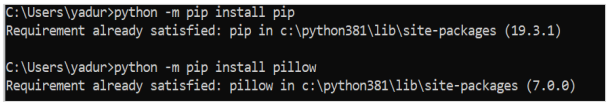
**Installing Pillow using pip**

To install pillow using pip, just run the below command in your command prompt:

python -m pip install pip

python -m pip install pillow

In case, if pip and pillow are already installed in your computer, above commands will simply mention the ‘requirement already satisfied’ as shown below:

****

**PYTHON PILLOW — USING IMAGE MODULE**

To display the image, pillow library is using an image class within it. The image module inside pillow package contains some important inbuilt functions like, load images or create new images, etc.

**Opening, rotating anddisplaying an image**

To load the image, we simply import the image module from the pillow and call the Image.open(), passing the image filename.

Instead of calling the Pillow module, we will call the PIL module as to make it backward compatible with an older module called Python Imaging Library (PIL). That’s why our code starts with “from PIL import Image” instead of “from Pillow import Image”.

Next, we’re going to load the image by calling the Image.open() function, which returns a value of the Image object data type. Any modification we make to the image object can be saved to an image file with the save() method. The image object we received using Image.open(), later can be used to resize, crop, draw or other image manipulation method calls on this Image object.

**Example**

Following example demonstrates the rotation of an image using python pillow:

from PIL import Image

#Open image using Image module

im = Image.open("images/cuba.jpg")

#Show actual Image im.show()

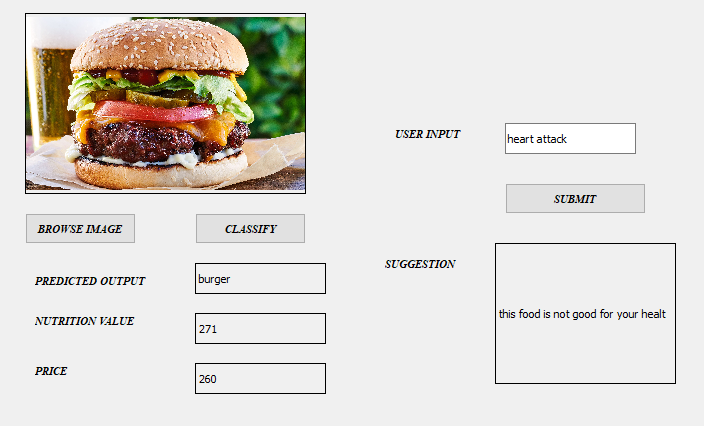
#Show rotated Image

im = im.rotate(45) im.show()

**Output**

If you save the above program as Example.py and execute, it displays the original and rotated images using standard PNG display utility, as follows:

**OUTPUT**



**RESULT AND DISCUSSION**

This section discusses the results, and the observations found while experimenting starting from the performance measurement techniques for food classiﬁcation.

The performance of the system is high, and is considered acceptable from a usage point of view. However, the CNN need high-performance computing machines in order to experiment on the huge multi-media datasets. The CNN is capable of train highly non-linear data, and for that in contrast, it takes more computational time to train the network.However, the performance matters a lot, and once the system is properly trained, the system can produce the results in less time. The images are properly pre processed and all kinds of images are tested with CNN. From this, it is concluded that CNN are more suitable for classifying the images when the number of classes are more.The task of image classiﬁcation can be extended using prominent features that can categorize food images. Since the CNN are consuming high computational time, the feature-based approach is highly appreciable. A multi-level classiﬁcation approach (hierarchical approach) is suitable to avoid mis-classiﬁcations when the number of classes is more.Moreover, a dataset containing all food categories is also not available in the literature yet.

**CONCLUSION**

Food plays an essential role in human life, providing various nutrients, and therefore the consumption of food is crucial for our health. Food classification is therefore a crucial aspect in maintaining a healthy lifestyle. In the world of health and medicine, food image classification is an emerging research area. A survey of automatic food classification methods based on Convolutional Neural Networks has been presented. The majority of the work uses the Food-101 dataset to train the models. Among the different approaches, InceptionV3-based systems provide higher accuracy in food image classification.

**REFERENCES**

[1] S. Yadav, Alpana and S. Chand, "Automated Food image Classification using Deep Learning approach," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 542-545, doi: 10.1109/ICACCS51430.2021.9441889.

[2] Abdulnaser Fakhrou, Jayakanth Kunhoth, Somaya Al Maadeed “Smartphone-based food recognition system using multiple deep CNN models”, Multimedia Tools and Applications volume 80, pages33011–33032 (2021)

[3] D. J. Attokaren, I. G. Fernandes, A. Sriram, Y. V. S. Murthy and S. G. Koolagudi, "Food classification from images using convolutional neural networks," TENCON 2017 - 2017 IEEE Region 10 Conference, 2017, pp. 2801-2806, doi: 10.1109/TENCON.2017.8228338.

1. Jiang, Malina. “Food Image Classification with Convolutional Neural Networks.” (2019). [5] Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Ma, Y. (2016).

[5] DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment. In: Chang, C., Chiari, L., Cao, Y., Jin, H., Mokhtari, M., Aloulou, H. (eds) Inclusive Smart Cities and Digital Health. ICOST 2016. Lecture Notes in Computer Science (), vol 9677. Springer, Cham. <https://doi.org/10.1007/978-3-319-> 39601-9\_4

[6] Vishwanath C Burkapalli, Priyadarshini C Patil, “Transfer Learning: Inception-V3 Based Custom Classification Approach For Food Images”, ICTACT Journal on Image and Video Processing ( Volume: 11 , Issue: 1 ),August 2020, doi: 10.21917/ijivp.2020.0321

1. M. T. Islam, B. M. N. Karim Siddique, S. Rahman and T. Jabid, "Food Image Classification with Convolutional Neural Network," 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2018, pp. 257-262, doi:10.1109/ICIIBMS.2018.8550005.

[8] Gözde Özsert Yiğit & B. Melis Özyildirim (2018) Comparison of convolutional neural network models forfood image classification, Journal of Information and Telecommunication,2:3, 347-357, DOI: 10.1080/24751839.2018.1446236

[9] Shen, Zhidong & Shehzad, Adnan & Chen, Si & Sun, Hui &

Liu, Jin. (2020). Machine Learning Based Approach on

Food Recognition and Nutrition Estimation. Procedia

Computer Science. 174. 448-453.

10.1016/j.procs.2020.06.113.

[10] S. Yadav, Alpana and S. Chand, "Automated Food image

Classification using Deep Learning approach," 2021 7th

International Co nference on Advanced Computing

and Communication Systems (ICACCS), 2021, pp. 542-

545, doi: 10.1109/ICACCS51430.2021.9441889.

[11] A. Arya and M. Manuel, "Intelligence Quotient

Classification from Human MRI Brain Images Using

Convolutional Neural Network," 2020 12th International

Conference on Computational Intelligence and

Communication Networks (CICN), 2020, pp. 75-80, doi:

10.1109/CICN49253.2020.9242552.

[12] Shipra Saxena, “Introduction to The Architecture of

Alexnet”, 2021

[13] https://neurohive.io/en/popular-networks/vgg16/

[14]https://medium.com/@mygreatlearning/what-isvgg16-introduction-to-vgg16-f2d63849f615

[15]https://blog.paperspace.com/popular-deep-learningarchitectures-resnet-inceptionv3-squeezenet/

[16] https://iq.opengenus.org/resnet50-architecture/

[17] https://iq.opengenus.org/inception-v3-modelarchitecture/

[18]https://www.analyticsvidhya.com/blog/2021/03/thearchitecture-of-lenet-5/

[19]https://www.datasciencecentral.com/lenet-5-a-classiccnn-architecture/

[20]https://towardsdatascience.com/review-squeezenetimage-classification-e7414825581a

[21]https://medium.com/coinmonks/paper-review-ofalexnet-caffenet-winner-in-ilsvrc-2012-imageclassification-b93598314160

[22]https://towardsdatascience.com/deep-learninggooglenet-explained-de8861c82765