IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IDENTIFICATION AND PREDICTION OF RECIPE USING DEEP LEARNING MODEL

[1]K Ranjith Reddy, [2]Jogi Krishna Mohan, [3]K Jayanth, [4]K Sagar

[1]Assistant Professor, [2,3,4]Student, Computer Science and Engineering, CMR Technical Campus, Hyderabad, Telangana.

Abstract: People enjoy food photography because they appreciate food. Behind each meal there is a story described in a complex recipe and, unfortunately, by simply looking at a food image we do not have access to its preparation process. Therefore, in this paper we introduce an inverse cooking system that recreates cooking recipes given food images. Our system predicts ingredients as sets by means of a novel architecture, modeling their dependencies without imposing any order, and then generates cooking instructions by attending to both image and its inferred ingredients simultaneously. We extensively evaluate the whole system on the large-scale Recipe1M dataset and show that (1) we improve performance w.r.t. previous baselines for ingredient prediction; (2) we are able to obtain high quality recipes by leveraging both image and ingredients; (3) our system is able to produce more compelling recipes than retrieval-based approaches according to human judgment.

***Keywords* - Inverse cooking, Recipe1M dataset.**

I. INTRODUCTION

Food is fundamental to human existence. Not only does it provide us with energy—it also defines our identity and culture. As the old saying goes, we are what we eat, and food related activities such as cooking, eating and talking about it take a significant portion of our daily life. Food culture has been spreading more than ever in the current digital era, with many people sharing pictures of food they are eating across social media. Querying Instagram for #food leads to at least 300M posts; similarly, searching for #foodie results in at least 100M posts, highlighting the unquestionable value that food has in our society. Moreover, eating patterns and cooking culture have been evolving over time. In the past, food was mostly prepared at home, but nowadays we frequently consume food prepared by thirdparties (e.g. takeaways, catering and restaurants). Thus, the access to detailed information about prepared food is limited and, as a consequence, it is hard to know precisely what we eat. Therefore, we argue that there is a need for inverse cooking systems, which are able to infer ingredients and cooking instructions from a prepared meal. The last few years have witnessed outstanding improvements in visual recognition tasks such as natural image classification, object detection and semantic segmentation. However, when comparing to natural image understanding, food recognition poses additional challenges, since food and its components have high intraclass variability and present heavy deformations that occur during the cooking process. Ingredients are frequently occluded in a cooked dish and come in a variety of colors, forms and textures. Further, visual ingredient detection requires high level reasoning and prior knowledge (e.g. cake will likely contain sugar and not salt, while croissant will presumably include butter). Hence, food recognition challenges current computer vision systems to go beyond the merely visible, and to incorporate prior knowledge to enable high-quality structured food preparation descriptions.Previous efforts on food understanding have mainly focused on food and ingredient categorization. However, a system for comprehensive visual food recognition should not only be able to recognize the type of meal or its ingredients, but also understand its preparation process. Traditionally, the image-to-recipe problem has been formulated as a retrieval task, where a recipe is retrieved from a fixed dataset based on the image similarity score in an embedding space. The performance of such systems highly depends on the dataset size and diversity, as well as on the quality of the learned embedding. Not surprisingly, these systems fail when a matching recipe for the image query does not exist in the static dataset. An alternative to overcome the dataset constraints of retrieval systems is to formulate the image-to-recipe problem as a conditional generation one.

II. IMPLEMENTATION

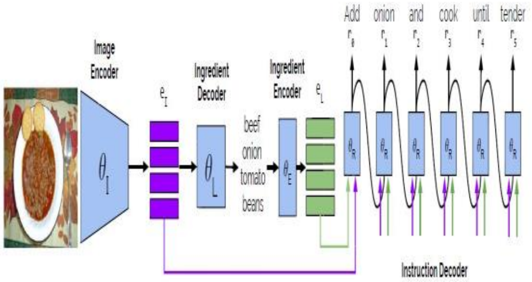
Previous efforts on food understanding have mainly focused on food and ingredient categorization. However, a system for comprehensive visual food recognition should not only be able to recognize the type of meal or its ingredients, but also understand its preparation process. Traditionally, the image-to-recipe problem has been formulated as a retrieval task where a recipe is retrieved from a fixed dataset based on the image similarity score in an embedding space. The performance of such systems highly depends on the dataset size and diversity, as well as on the quality of the learned embedding. Not surprisingly, these systems fail when a matching recipe for the image query does not exist in the static data.

Disadvantages of existing system:

However, a system for comprehensive visual food recognition should not only be able to recognize the type of meal or its ingredients, but also understand its preparation process. In this project we are training CNN with recipe details and images and this model can be used to predict recipe by uploading related images and we used 1 million recipe dataset and from this dataset we used 1000 recipes as training entire dataset with images will take lots of memory and hours of time train CNN model.

Advantages of proposed system:

The contributions of this paper can be summarized as: We present an inverse cooking system, which generates cooking instructions conditioned on an image and its ingredients, exploring different attention strategies to reason about both modalities simultaneously. We exhaustively study ingredients as both a list and a set, and propose a new architecture for ingredient prediction that exploits co-dependencies among ingredients without imposing order. By means of a user study we show that ingredient prediction is indeed a difficult task and demonstrate the superiority of our proposed system against image-to recipe retrieval approaches.



To implement this project we have designed following modules

1) Upload Recipe Dataset: Using this module we will upload dataset to application and then read all images and recipes details and then store them in array

2) Build CNN Model: Using this model we will entire recipe array and then input those details to CNN model to train CNN on recipe dataset

3) Upload Image & Predict Recipes: using this module we will upload test image and the application will predict recipe for that image.

III. MODULES

The project has been classified into six modules (or stages) in a sequential order that the Input image has to go through to remodel itself into a valuable Net numeric Score that proves to be an asset in commercial business. This modular approach of the project is shown below sequentially.

Module 1: Data Cleaning

Besides unraveling the hidden patterns and insights, Data Exploration allows one to take up initial steps in building a distinctly accurate model.

- Major time needs to be spent on data exploration, cleaning, and preparation as this would eat away a crucial portion of project time.
- It also supports superior and well-curated analysis and improved business intelligence for processing and decision making.

Although our core_data_recpie.csv dataset includes 1125 Multi-cuisine recipes, but still a lot of them cannot be used due to lack of appropriate quality. Many community cooking blogs contain multiple recipes of a single food dish that are largely unstructured. As part of the Data Cleaning pipeline, we have identified few images and textual information and cleaned them as follow:

Instructions: Manual Investigation of several records in the given dataset, proved some users have given URLs or emoticons and other special symbols in the cooking instructions text. The distribution of Instruction length was observed to be Right skewed (i.e., Instruction length in terms of the number of characters that fall in the positive region).

Ingredients: The ingredient list makes up most of the unstructured portion of the dataset. It consists of roughly around 10,000 unique Ingredients mainly because they also contain pronouns and adjectives (e.g., Turkey Black Bean Burgers instead of only Bean Burgers).

- **Removal or replacement of Special symbols** (such as @, ! , -, *,) with blank spaces.
- **Handling Compound words:** When two or more **words** are concatenated to form a new **word** that has a completely different definition. Such words are called Compound words.

Image Scrapping: Image scraper used certain requests library to extract food images from the specified websites. The request libraries involve Beautiful Soup and pandas to export scraped data (i.e. image URLs) and present output data into our core_data_recipe.csv file. We assigned an appropriate web driver to pick the URL from which we scraped image links and created a list data structure for storing.

Image Resolution: For training an accurate model, at least four food images of the same dish with adequate resolutions are necessary. Recipes without corresponding images were also retained as they can be matched with the ones with images. We have resized the food images to 63*63*3 pixels for our model input.

Module 2: Data Pre-Processing :

Importing Libraries: We begin with importing the libraries that would be required to perform certain tasks in the code. Library is basically a set of built-in modules by the developer during the installation which can be called and used whenever required within the program.

```
import numpy as np
```

```
import pandas as pd
```

Splitting the Dataset: The entire dataset is split into Training Set, and Validation Set, and Test Set of similar distribution. A 60-20-20 split was performed on the existing dataset into three categories as follows:

- **Training Set:** This contains 60% of the entire dataset, which is required used to train the CNN model.
- **Test Set:** This contains 20% of the entire dataset, which is used in estimating the performance of the model in terms of accuracy and loss.
- **Validation Set:** This contains 20% of the entire data, which is used to fine-tune the hyperparameters to determine the model learning rate.

Split count of Food Images:

- Count of Training images: 1000
- Count of Test images: 125
- Count of Validation images: 20

Input pipeline: An Input pipeline that is nothing but the core_data_recipe dataset, loads the training examples in the form of a **tuple (n, img x(ing), x(inst))** to the model.

- Here n is the unique recpie_number.
- img represents the food images in the dataset.
- x(ing) denotes the Ingredients list and;
- x(inst) has the cooking instructions along with timings in the form of text.

Images: The cleaned dataset includes 4 images of each food item. During the training phase, an image is selected randomly from the remaining 3 images for better learning. A lesser training set leads to Overfitting (i.e.: For instance, the model recognizes food images in the training data. So, the accuracy of the training set will be greater than that of the Test set which is termed as overfitting).

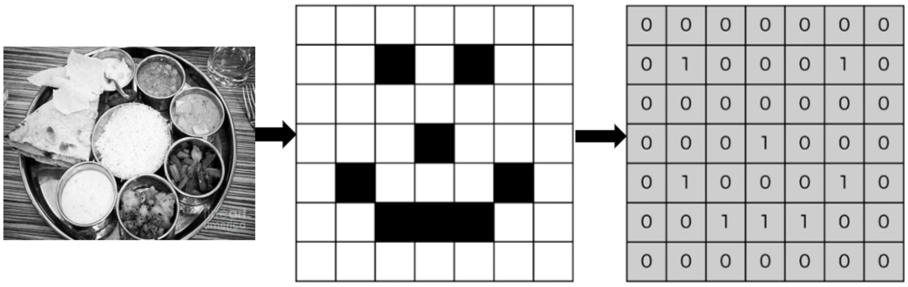
Ingredients: Ingredients are tokenized by using a word encoder with fixed-sized vocabulary. Breaking a character string into pieces, called Tokens, and throwing away certain characters at the same time, including punctuation and special characters. A special delimiter '^' is used to separate ingredients of the ingredients list in the data set. The ingredients in the ingredients are placed in a random order as the order does not affect the model learning.

Title and Cooking Instructions: The recipe title is generated using the major ingredients in the ingredients list. We have concatenated the title and the cooking instructions. Cooking instructions also include the time required to perform every instruction and the total time becomes the sum of such time slots.

Module 3: Building Cnn Model

A renowned Deep learning class that is employed Image Recognition is CNN. **Inspiration of VGGNet:** The most preferred VGGNet was designed by Visual Geometry Group and stood as the runner-up in the ImageNet competition as the most effective Feature extraction model. Inspired from the VGG-16 model which is 16 hidden layers and pre-trained on more than one million images of around 1000 objects from the ImageNet database, we have built a model of the same architecture for ingredient recognition for the food image.

Implementation CNN Architecture in Reverse Cooking: Every image consists of pixels ranging from 0 to 255 and three color channels namely- Red, Green, and Blue (RGB color pallet). So, to simply we made this black and white image with 0 and 1 pixels.



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Step 1: Convolution- Here we have taken an image of 63*63*30 pixels and for feature detection, we make use of a filter of size 4*4*4. The filter is rolled throughout the image and a cartesian product is performed to generate a feature map. This is repeated until the entire image is featured in the feature map by moving the filter horizontally and vertically. After creating feature maps, we arrange them together to form our first convolutional layer.

Step 2: Max Pooling - Upon obtaining the Convolution layer, each feature is taken and Max pooling is done. **Max pooling** is a **pooling** function that chooses the **maximum** element from the feature map covered by the filter of size 4*4*4. After performing max pooling on all the features map, we obtain a matrix with all the maximum elements. Such a matrix is called a Pooled Feature Map which highlights the most interesting and present feature in the food image. Pile of Pooled Feature Map makes up the Pooling Layer. This layer has only the most significant features of the food image which paves way for quicker computations.

Step 3: Flattening- Conversion of the obtained Pooled Feature map data into a 1-D array for inputting it to the next subsequent layer for faster processing is called Flattening. A flattened layer is nothing but an exclusive long feature vector vertically. The flattened layer is connected to the CNN model, which is the *fully connected* layer. The output of the flattened layer is used by the Dense layer of the subsequent neural network model.

Step 4: Full Connection- Fully connected layers are inputted with Pooled features map obtained during Pooling, which undergoes flattening before being inputted to a fully connected layer. CNN Model consists of three distinct layers namely:

- **Input layer** which comprises of Input data.
- **Hidden layers** which include activation nodes called neurons; and
- An **Output layer** includes one or more neurons, whose outputs are the final output of the network.

Unlike the Input layer and hidden layers which use the ReLU activation function, the SoftMax activation function is employed in the output layer. SoftMax to obtain class probabilities of the input for *classification. Whereas* **Re**LU is a max **function performed on the neurons in the layer to give only** positive and zero values.

Translation of features into another form in Sequence transduction models is quite challenging since the input and output are both variable-length sequences. Such types of models are handled with the help of Encoder and Decoder sequential architecture. The Reverse cooking model also has 1 encoder- Image encoder and 2 decoders namely: Ingredient decoder and cooking Instruction decoder.

Module 4: Image Encoder

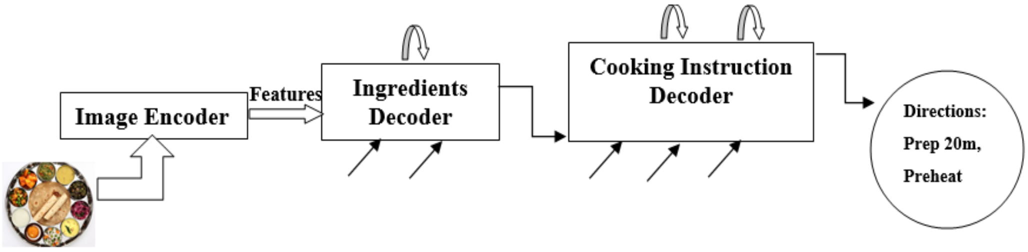
Generally, Encoder takes a variable-length input sequence and then transforms it into fixed-size output. Here, the image encoder maps an input image img into a feature representation X which consists of the number of image features and the embedding dimension. Upon image data encoded to an integer, the embedding layer represents each word in the form of a unique numeric value. Typically, the decoder maps the encoded output of a fixed length to a variable-length sequentially by taking inputs and generating outputs at each intervals of time.

Module 5: Ingredients Decoder

It is a transformer decoder network that is conditioned on the image features x to produce feature vectors denoting the number of ingredients. The decoder output of the will be inputted to a linear output layer followed by a SoftMax activation function to generate the predictions. SoftMax activation ensures that the output probabilities sum up to 1.

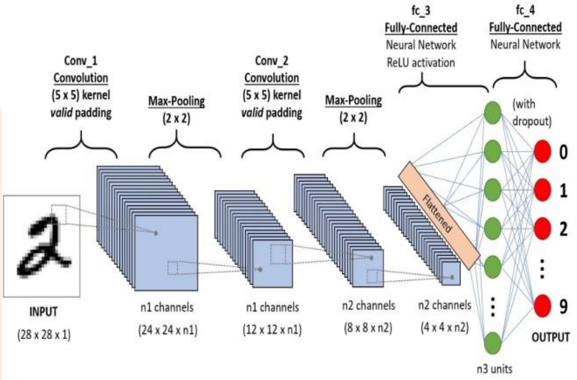
Module 6: Cooking Instructions Decoder

The output of the Ingredients decoder is given to the cooking instruction decoder for the generation of food recipes. It is conditioned on both the embedded image img and the ingredients features x obtained from the Ingredients decoder. Upon processing, the output of the Instruction decoder was fed into a linear layer followed by the SoftMax activation function to generate probabilities over the vocabulary of cooking instructions text.



IV. CNN Used in Deep Learning

Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. Researchers and enthusiasts alike, work on numerous aspects of the field to make amazing things happen. One of many such areas is the domain of Computer Vision. The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm — a Convolutional Neural Network.



A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space. Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

.

V. Acknowledgment

We thank CMR Technical Campus for supporting this paper titled with "IDENTIFCATION AND PREDICTION OF RECIPE USING DEEP LEARNING MODEL", which provided good facilities and support to accomplish our work. Sincerely thank to our Chairman, Director, Deans, Guide and faculty members for giving valuable suggestions and guidance in every aspect of our work.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In ECCV, 2014.

[2] Micael Carvalho, R´emi Cad`ene, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In SIGIR, 2018.

[3] Jing-Jing Chen and Chong-Wah Ngo. Deep-based ingredient recognition for cooking recipe retrieval. In ACM Multimedia. ACM, 2016.

[4] Jing-Jing Chen, Chong-Wah Ngo, and Tat-Seng Chua. Cross-modal recipe retrieval with rich food attributes. In ACM Multimedia. ACM, 2017.

[5] Mei-Yun Chen, Yung-Hsiang Yang, Chia-Ju Ho, Shih-Han Wang, Shane-Ming Liu, Eugene Chang, Che-Hua Yeh, and Ming Ouhyoung. Automatic chinese food identification and quantity estimation. In SIGGRAPH Asia 2012 Technical Briefs, 2012.

[6] Xin Chen, Hua Zhou, and Liang Diao. Chinesefoodnet: A large-scale image dataset for chinese food recognition. CoRR, abs/1705.02743, 2017.

[7] Bo Dai, Dahua Lin, Raquel Urtasun, and Sanja Fidler. Towards diverse and natural image descriptions via a conditional gan. ICCV, 2017.

[8] Krzysztof Dembczy´nski, Weiwei Cheng, and Eyke H¨ullermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In ICML, 2010.

[9] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In ACL, 2018.

[10] Claude Fischler. Food, self and identity. Information (International Social Science Council), 1988.