ABSTRACT

Briefly outline your project idea. What will be the inputs (e.g., animal images or bird sounds) and input method (e.g., upload to a webpage, take a photo w/ a cellphone, etc.)? What will you output (e.g., the Wikipedia article about the animal or bird) and how will you deliver this output (display on the webpage/phone, spoken, etc.)? What ML algorithms/model types will you use for the ML part of the task?

The project idea is to combine two non-ML elements with ML elements to make a simple fruit-vegetable cooking guide. With this cooking guide, users could use their own device's built-in camera to shoot the raw materials they want to make (users do not need to know their names), or they could directly upload prepared pictures. Then, they could get several different cooking methods (recipes). Briefly, the inputs are photos of fruits or vegetables users want to cook. One input method is taking photos by the built-in camera, another is uploading existing photos. After identification, the name, the brief introduction, and some cooking methods of the fruit or the vegetable in the photo will be displayed. The brief introduction will be from Wikipedia. The cooking methods will show up as YouTube video links or links to cooking articles. If the input is not one kind of fruit or vegetables, then the guide will tell it's neither a fruit or a vegetable. All operations will be completed by computers, hence the outputs will also show on the web-page. For the ML part of the task, MobileNet would be considered, which puts forward the concept of depthwise separable convolution, and does well in reducing computational costs.

What specific ML architecture do you plan to use to solve the ML task? (This can be an existing architecture like ResNeXt or BERT, or you could briefly describe your model architecture) If you intend to do transfer learning, where do you plan to get the pre-trained model from, and what was it trained on originally? Be specific: saying you will use a CNN for a visual task is way too broad.

For the ML part of the task, MobileNet would be considered. The basic unit of MobileNet is depthwise separable convolution, which is a factorized convolution, which can be divided into two smaller operations: depthwise convolution and pointwise convolution. The overall effect is similar to that of a standard convolution, but it will greatly reduce the amount of computation and the number of model parameters. For transfer learning, the official pre-trained models could be considered (Github for pretrained MobileNet models: https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet). There are multiple models with different input sizes, different numbers of channels in different networks we could see from this Github, and the corresponding accuracy for each model is provided. These models have been trained on the ILSVRC-2012-CLS image classification dataset. The project could choose a suitable size model for transfer learning, according to the expected size, resolution of the input image.

DATA AND SOURCES

What data will you train/retrain from? How many samples do you estimate you will need? Where will you collect the data and labels from, and how? If you plan to combine several data sources to create your dataset(s), please describe all of the sources. Be specific: saying that you will find images on the internet is not enough, you have to specify a specific source that you verified has the kind of data you need.

The data used to train/retain would be from the combination of different fruit-vegetable datasets and some manually collected images. This project plans to use at least a 110k size combined dataset (Train: Test is around 8:2). Some existing datasets would be collected from some public Kaggle and websites. One is a dataset called Fruit 360 dataset with 90483 images of 131 fruits and vegetables
(https://www.kaggle.com/moltean/fruits). And there is a good introduction for the Fruit 360

dataset (https://levelup.gitconnected.com/fruits-vegetables-and-deep-learning-c5814c59fcc9). Another is a more simple dataset about fruit and vegetable (https://www.kaggle.com/kritikseth/fruit-and-vegetable-image-recognition). Manually collected images would be collected from taking photos of vegetables and fruits and downloading pictures of fruits and vegetables from websites, using Bing tools, e.g. Bulk Bing Image downloader (https://github.com/ostrolucky/Bulk-Bing-Image-downloader). To obtain the final dataset the project will use, the Fruit 360 dataset will be the core to consolidate all the data. The folders in the Fruit 360 will keep essentially unchanged. Other pictures will be divided into the corresponding folder (which has the name of vegetable or fruit). If there are some types that Fruit 360 does not have, corresponding folders with the category name will be created. The numbering methods for all folders will be uniform.

NON-ML COMPONENT

How will you implement the non-ML component? What will it run on (web, phone, desktop, etc.)? For example, if this will be a dynamic web service, where will you host it, and what languages/libraries will you use to implement the web backend?

For the input part, Vue. js which is a progressive framework for JavaScript could be used to build web interfaces that the project required. The necessities for this web interface are two bottoms: “take a photo” and “local upload”. Then use Jave to build a mapping relation between the fruit-vegetable and corresponding introduction, cooking methods. Also, use Jave to implement the web backend. One way is to use a fixed computer as the backend for the entire project. Another way is to build up a website, then it will require a domain name, IP address, cloud server (google cloud platform could be considered), and so on to achieve the whole web backend.

EXISTING SIMILAR WORK

Are there any websites/apps/projects that do something very similar to what you propose? If so, cite them, and briefly explain how your project will differ. Also, cite any work that you expect to reuse/adapt.

Some cooking teaching websites have similar features to our proposal, they will also provide several cooking methods to help viewers make delicious food. The users of these sites usually know simple information about fruits and vegetables. However, our project is also suitable for complete beginners. Users are not required to know the name and related common sense about the material they plan to cook when they use our fruit-vegetable guide. The second difference is that users learn the recipe first then come up with the ingredients with previous sites, while our project can target a particular type of fruit and vegetable that users most want to use.
There are some websites we will use to find recipes:

1. https://www.epicurious.com/ (a website includes many recipes, users could find recipes by entering the keywords)
2. https://www.cookingchanneltv.com/recipes (Similar to the first one)
3. https://www.delish.com/ (users could find the recipes they want through the classified part)
4. https://www.simplyrecipes.com/
5. https://youtube.com/ (could find cooking videos by searching keywords)

WORK DISTRIBUTION

In the dataset collection stage, one person takes charge of collecting images of fruits and vegetables, the existing datasets of fruits and vegetables, and combining this information, one person collects a brief description of each fruit and vegetable, and two other people are responsible for collecting recipes (which can be, but are not limited to, YouTube videos).

For two Non-ML components, two people are responsible for one: two people are responsible for the web front-end, which is the page that users use to input. The other two are responsible for the web backend with mapping relation between fruit-vegetable and recipes.

For the ML part, four people will learn MobileNet together, two people completed one, and after training of the same order of magnitude, the one with higher accuracy will be selected for subsequent optimization. For training and testing stages, choose a computer with better performance to save time.

MILESTONES

The four milestones point to the three components (two non-ML, one ML) and the final integration stage respectively. The first milestone is the collection step, where group members will do background research into how MobileNet works, and how to implement the web app for the project. The second milestone will have a basic web setup to take in images from either a camera or a folder on the computer, and have at least an 80% accuracy for the model. By the third milestone, the webpage should be able to output recipes after taking in an image, and the model should reach around an accuracy of 90%. The last one is to handle back-end logic, coordinate three components, achieve the final goal (input picture then output name, introduction, and recipes).

RISKS AND BACKUP PLAN

What are the key risks in your project, and what is your fallback plan if the risk occurs? We are perfectly fine with you taking big risks — that is part of education — but we want to know you won't be left with no options if something goes pear-shaped.

The first risk is the applicability of public datasets since not all the common fruits and vegetables will appear in the existing datasets. If this situation exists, there are two possible ways to handle it. One is to supplement images manually. Another is to print “no result” to users at first but design a suggestion box for users to collect their demands to improve our dataset. Here, it should be mentioned that the Fruit 360 is a dataset for image classification, all images in it are of a single object (e.g. a single apple, a single banana). However, collected images will usually contain more than one item. Thus, integration of datasets may require separating these two situations. The second risk is MobileNet is one of the image classification models with no object detection, if there are multiple detection targets in the picture, the accuracy of the test will be greatly reduced. The possible ways to solve it are two: one is to mention the input format at the user interface, another is to add an object detection function from some object detection model such as yolo3. Another risk needs to be mentioned is that many fruits have very similar appearance, such as orange, clementine and kumquats. The third risk is that our MobileNet might not reach the expected accuracy since it is one of the lightweight network models. Although this kind of deep separable convolution can reduce the amount of calculation, it will also lose some accuracy. The fallback plan is to improve MobileNet by adding data, adjust data augmentation, adjust the learning rate for training, modify the size of the preset box to make it closer to the actual size of the object. Also, adding residual blocks of ResNet to MobileNet might be a good try if previous steps do not have expected effects. The fourth risk is that the work for finding corresponding recipes is huge while designing a smart search program, which could select and collect suitable web pages based on keyword search, is a good method.