Examining Computer Vision from a theoretical perspective: improving YouCook and creating YouCookWeb

Ariel Tello
Computer Science Department
University of Rochester
atello@u.rochester.edu

Abstract

Computer Vision researchers have extensively explored how to use images in technology like facial recognition, image segmentation, and image classification. Yet, have limited research when it comes to using videos as visual data input because of its time and space expensive nature. YouCook's focus is to use instructional videos to enable a machine to develop reasoning, so that eventually when given any video it can understand the video's content and its relationship outside the video's context. In order, for a machine to learn such a skill it needs to train on a quality dataset. The refined model for YouCook has steered away from using just the QA approach to incorporating dependency relationships of content within video segments. The idea is to create a relational prediction network so that the machine can eventually generate graphs which would link together concepts from the video, thereby demonstrating reasoning and comprehension. To collect data and form this relational prediction network, YouCookWeb was created for human data annotators. YouCookWeb is a platform for these annotators to mark dependency relationships between video segments in an intuitive way which group together data for the researchers to analyze. By analyzing the data, patterns can be identified and the numbers behind training the model (weights, bias values, probabilities, etc…) can be modified to improve the machine's prediction algorithm.

1. Background

What is Computer Vision?

For those unfamiliar with the term "Computer Vision," it may sound complicated, but it doesn't mean it can't be broken down into simpler words. Essentially, Computer Vision is a field in Computer Science that focuses on training a computer to see, process, and identify the visual world. Images, videos, and deep learning models aide training the computer to learn and produce the expected output. For example, when you take a picture of a paper check through a bank app, the way it deposits the check and verifies information uses Computer Vision! The scanned image applies OCR (Optical Character Recognition) to "read" the handwritten text in order to verify the information and deposit your money.

Now you might have heard terms like Artificial Intelligence, Machine Learning and even Deep Learning tossed around. These are not synonymous words, but they do have a relationship with one another. If we were to think of it as a hierarchy (check Figure 1) then Artificial Intelligence encompasses all, then we have Machine Learning, and within Machine Learning there is Deep Learning. Here's a quick-term reference if you're not sure of their definitions.

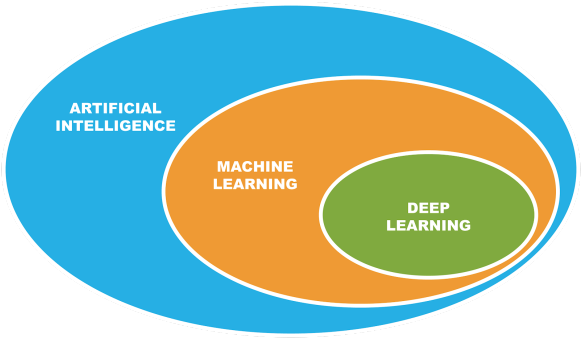


Figure 1: This shows the relationship between AI, ML and DL.

Definitions:

- Artificial Intelligence (AI) is what has enabled machines to become intelligent so they can think and work like humans like problem-solving (ex: Siri, spam filters, etc...).

- Machine Learning (ML) is a branch in AI that uses minimal human intervention for systems to learn data, identify patterns, and other applications.
- Deep Learning (DL) subset of ML that uses neural network models to enable learning from the dataset in a supervised (labeled data) or unsupervised (data with no labeled responses) manner.

What is a Neural Network?

Now that you have a general idea of what Computer Vision is, it's important to recognize that Computer Vision has now shifted its approach to incorporate Deep Learning techniques. Neural networks are what allows the machine to "learn" concepts in order to perform certain tasks, and it is also modeled after the human brain's neurons hence its name. How a neural network works is a long process to explain, but this is a brief explanation of its functionality.

How does a Neural Network work?

A common depiction of a neural network is Figure 1.1, you have 3 layers that are apparent. The input layer takes in the data, the hidden layer processes the data, then the output layer produces predictions based on the inputs. The circles are actually nodes, and in between them, they have connections that contain a weight which affects the node in the next layer. As the input is processed, the node values are manipulated and that manipulated value will then be given to the top hidden layer node which says whether it is activated or not. Its activation then determines whether a bias node is used, this bias node is important for the model because it shifts the activation function which makes for better predictions (outputs). You could see all of this in Figure 1.2. Without getting too technical, the model trains for many epochs and after each run, it changes values like the weight and bias values, in order to optimize the sum of errors (cost function) which is performed to make the model more accurate.

It is important to note that once the model trains on a large dataset that is usually split into 3 separate and unique datasets. The first dataset is for training, the second is for validation, and the third is for testing. The second dataset means the training dataset surpassed the baseline result which is good. The testing dataset would be the true determinant of whether or not the machine has truly learned the objective.

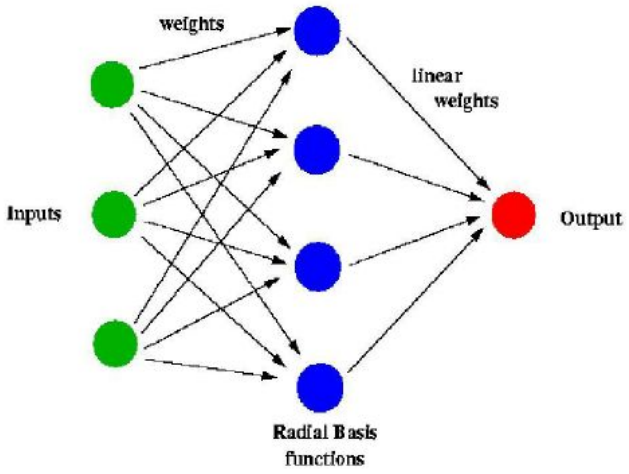


Figure 1.1: basic neural network with 3 layers, nodes, and "connections"/weights

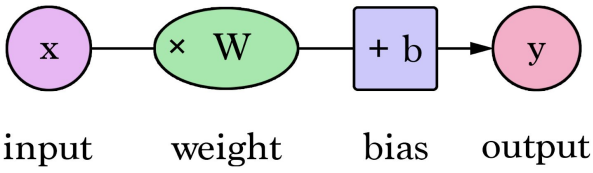


Figure 1.2: the basics about the mathematical part of computing predictions of a deep learning model

What is YouCook?

The project of interest is YouCook, which is an annotated QA dataset (check Figure 1.3) for instructional videos. The primary goal of this project is to find a way to teach a machine to reason through these instructional videos. Videos are ubiquitous, yet there isn't enough research studying how to enable a machine to learn with these long sequences of data. The idea is to use instructional videos since it is structured and these videos are step-by-step procedures making it constrained to just an understanding task. The way the model learns is via a supervised learning technique, it uses a Question - Answering dataset on multiple segments of videos.

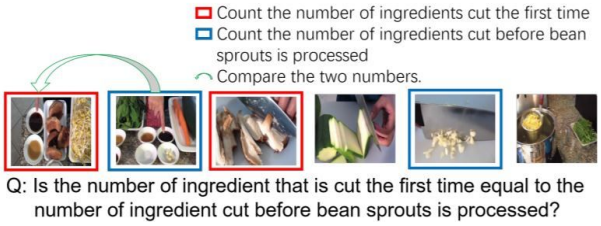


Figure 1.3: example of how annotators curated a question based on video segments given

The current dataset has over 15k questions and over 75k possible answers for those questions. This dataset has questions asking about the number of objects, actions, the taste of ingredients, order of actions or ingredients, as well as, ones involving time. Example questions are: When was x added? Is y salty? How many of x or y colored objects are used in the recipe? Although this dataset is one of the first of its kind, oriented for instructional videos, it has shortcomings that will be discussed in the "2. QA Dataset" section of this paper.


```
{
  "question": "Is cheese the only topping of the bread?",
  "answer": "yes",
  "alternatives": [
    "no",
    "tomatoes",
    "7",
    "65"
  ],
  "type": [
    4,
    0
  ]
},
{
  "question": "When is the sandwich flipped?",
  "answer": "140",
  "alternatives": [
    "169",
    "117",
    "123",
    "98"
```


Figure 1.4: example questions and answers from the original dataset

Admittedly, this paper is lengthy, but it is conclusive for those familiar in the field and informative for those who don't know about this field at all. The rest of the paper is organized in the following way.

- Sec. 1.2 - discusses related works to video understanding through notes taken on academic papers
- Sec. 2 - delves into the QA dataset of YouCook and its shortcomings
- Sec 2.1 - explains my contributions to the QA dataset
- Sec. 3 - discusses the new approach to incorporate a relational prediction network to hopefully improve the model's prediction accuracy
- Sec. 3.1 - gives links for related works using deep learning models implementing graphs
- Sec. 4 - describes a new model for this project
- Sec. 5 - explains YouCookWeb and how it facilitates data collection and analysis
- Sec. 6 - discusses future applications of YouCook and how image/video research in this field impacts technology, as well as, affects us in general

1.2. Related Work

TVQA dataset:
https://www.aclweb.org/anthology/D18-1167
- data limitations = less work on video-based QA
- TVQA = 152k QA pairs from 21.7k clips for 460 hours of video
- relevant moments in clip + dialogue + visual concept recognition = ability to answer question
- What do the models actually comprehend?
- VQA systems take images or videos as input w/ natural language questions to make answers to those questions
- by asking a combo of object identification, counting, appearance, to more complex q's = better idea of model's semantic understanding
- shortcomings are common for video QA datasets out there
- QA pairs are written by people observing videos and dialogues, want to make questions that require both vision and language comprehension
- most important area of study may be the associated natural language of subtitle dialogue to movies = reflects real world
- dataset built on popular TV shows, has 4 advantages
  - ~ large-scale and natural
  - ~ relatively long clips
  - ~ dialogue (name+subtitle content)
  - ~ questions are compositional algorithms need to find relevant moments
- moment localization = localize a short moment from a long video sequence given a query description helps answer compositional questions
- compositional format ==> [What/How/Where/Why/..] ____ [when/before/after]_____.
- proposed 2 QA tasks (w + w/o timestamps) to provide a baseline
- the experiment was to test both visual and textual understanding

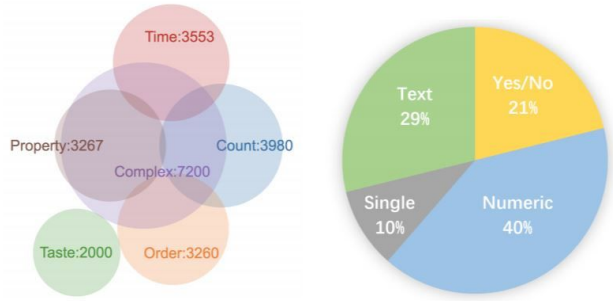
Temporal Relational reasoning in videos

https://eccv2018.org/openaccess/content_ECCV_2018/papers/Bolei_Zhou_Temporal_Relational_Reasoning_ECCV_2018_paper.pdf
- temporal relational reasoning = ability to link meaningful transformations of objects or entities over time
- TRN (temporal relation network) = learn+reason about temporal dependencies between video frames at multiple time scales
- ability to reason about relations between entities over time is important for decision-making; analyze current situation relative to the past and make a hypothesis
-video datasets like UCF101, Sport1M, THUMOS have activities that can be identified without reasoning
-CNN struggle where data is limited; if structure has undergone transformations and temporal relations
- not modeled after spatial relations but by temporal relations between observations in videos; so TRN can learn and discover possible temporal relations at multiple time scales

YouCook gcn paper
https://arxiv.org/pdf/1812.00344.pdf
- existing video methods focus on short-term actions (not applicable for videos of varying length)
- instructional videos make a task harder to understand
- YouQuek = an annotated QA dataset for instructional videos (YouCook2)
    ~related to logical reasoning in the temporal dimension (not exclusive)
- RGCN = Recurrent Graph Convolutional Network captures temporal order and relation information
    ~performs the best for QA accuracy and with the inclusion of human annotated description
- can machines also understand instructional video as humans?
    ~need accurate recognition of objects, actions, and events + higher-order inference of any relations therein
- question-answering (QA) task in instructional videos acts as a proxy to benchmark higher-order inference in machine intelligence
- YouQuek Dataset
    ~Counting
    ~Time
    ~Order
    ~Taste
    ~Complex
    ~Property
-Model:
    - pre processing procedure
    - temporal boundaries = human annotated start/end time stamps of procedure
    - segments (procedures) by temporal boundaries + descriptions also added by humans
    - transcripts are auto generated by speech recognition

2. QA Dataset

The QA pairs had a series of questions for a video segment and each question has 5 possible answers. The QA dataset as a whole was categorized into 6 types of questions. They are labeled as counting, time, order, taste, complex, and property. The statistics of the questions are demonstrated in Figure 2. Upon investigation, it was seen that the pairs had low diversity which means it will not be an effective dataset. An ineffective dataset (check Figure 2.1 for an example) also means that biases are most likely present and it could be the reason why the baseline result was a 33%, which was not high enough to pass into the validation or the testing dataset.



(a) Distribution on categories. (b) Different answer tags.

Figure 2: statistics on the original QA dataset given by the YouCook gcn paper

There were biases associated with the dataset like the fact that all the possible answers were not related to one another. For example, possible answers to a question could be “yes”, “no”, “1”, “203”, “Texas”. To a human, these answers seem preposterous since they don’t seem to be at all related to one another. To a machine, this means there is a pattern to these questions. If the options include “yes” and “no” then it must be a yes/no question so it has a 50%

chance of choosing the right answer. This creates a bias because at this point it is not learning, but just learning how to guess the correct answer. Another popular question in the dataset were ones consisting of what color is x ingredient. If the answer to that question was that the bread is brown after toasting, then it will keep guessing that answer for the future question which is again a learning pattern.

Not to mention, the questions are not representative of complex questions. “Is an egg added into the dough mixture?” is not a question that an abundance of new information can be learned from. The goal of the project is not to be an image classification program. We don’t want to know if what is present in the clip is an egg, but why is an egg used in that clip. That mapping of concepts outside the context of the video is the goal, it requires high order inference and reasoning to determine why an egg is used in a clip. Likewise, another question that is not as complex is time questions. When a question consists of when something was added, it is only finding the object of interest and that is object identification and doesn’t help further build the machine’s knowledge of why it is used.

So, the questions in the dataset labeled as counting, time, and taste should be used as a set for object identification. This set would be helpful for the algorithm to learn what object is what and then after it recognizes the object it can label relationships between other objects. This set should be separate than the objective of the project though, so questions that are identifiable by action, order, and property should be used mainly for this dataset. These sorts of questions are not only more representative of human logic, but they are truly complex and do allow a machine to learn. If you can identify objects, know its order of addition or manipulation, as well as, its property change you will surely learn the step-by-step procedure nature of these sorts of videos.

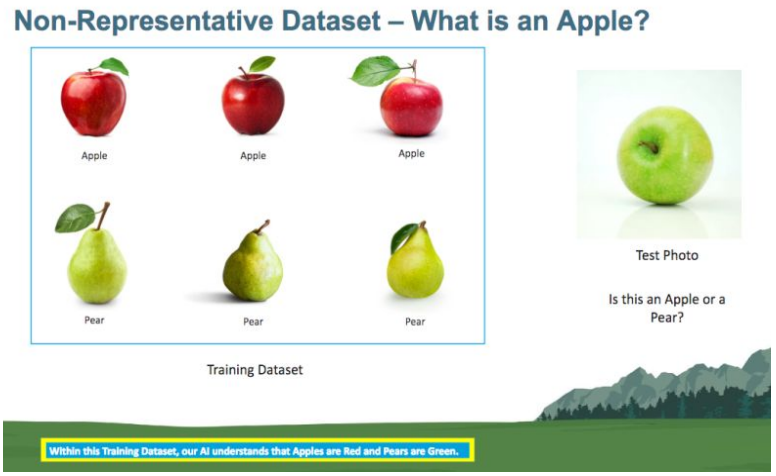


Figure 2.1: a simple depiction of a non-representative dataset

2.1. My QA Dataset Contribution

The QA pairs that I contributed to the dataset had a different composition than the original. My contribution consisted of over 500 questions and over 2.5k answers which is not nearly enough for a complete dataset. For every video, there was a minimum of curating 7 questions and each question must have 5 possible answers. Each answer was a complete sentence and not one-word answers which made for a time-consuming data annotation process.

When the dataset was visualized in Figure 2.2, it was found that most of the frequent words were question words which is how the dataset should be. In the original dataset, it was seen that words like “yes”, “no”, “time” and irrelevant words were common which could only mean the machine is not taking as much information as it could if the questions and answers were curated with more care.

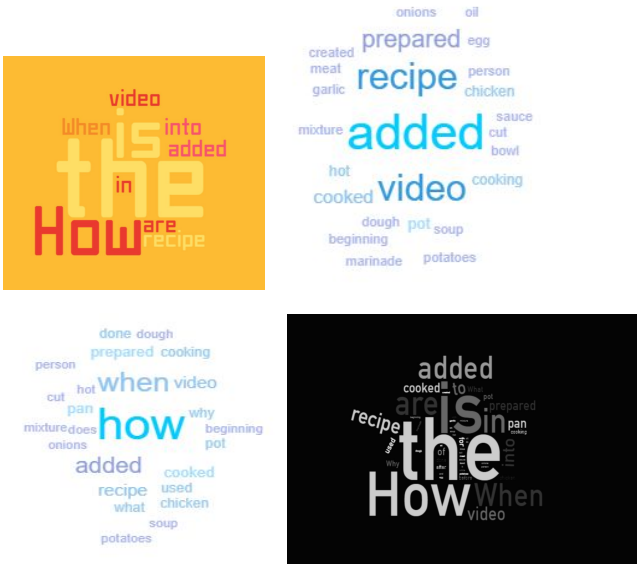


Figure 2.2: data visualization of my QA dataset

3. Relational Prediction Network

The relational prediction network that we want to create would be helpful for the eventual goal of graph generation with high prediction accuracy. We want to link concepts together like (Figure 3.1) an egg with cooking, its different preparation methods, or like a pan because that is how it cooks. We believe that creating such a network will allow the machine to formulate relationships between similar objects or concepts in video segments.

If the machine is able to come up with such relationships then it can potentially link together more abstract concepts which is ideal for video understanding. Thus, this network would be used for future datasets and data from it could be extracted to find other features the model can train on.

I think this is a novel idea that can be used in other ways as well such as creating a sort of map of concepts that can be given more details and nodes by human annotators. Then if a concept or object is in the video it can follow a predetermined pathway (similar idea to Figure 3.2) that has knowledge and form new connections between other concepts it learns about. As you can see this can be applicable in various ways and it would be interesting to see another researcher to able to develop this idea even further.

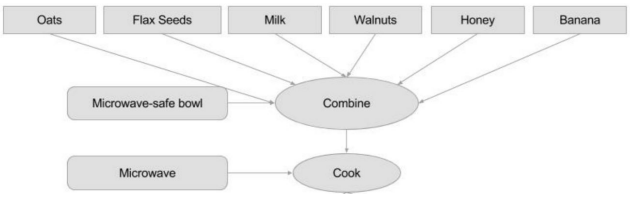


Figure 3.1: an example of video concepts of the videos linked together; graph generation would consist of video segments in YouCookWeb

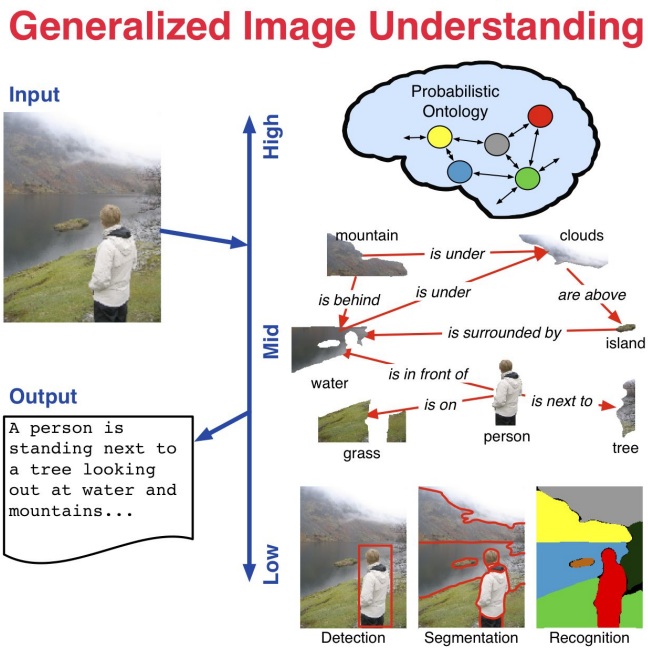


Figure 3.2: Jason Corso at the University of Michigan also works with video understanding and this picture explains a similar idea that could potentially work as a baseline model as well

3.1. Related Work

[1]https://www-cs.stanford.edu/people/jure/pubs/graphrepresentation-ieee17.pdf
Representation Learning on Graphs: Methods and Applications

[2]https://arxiv.org/pdf/1812.08434.pdf
Graph Neural Networks: A Review of Methods and Applications**

[3]http://tkipf.github.io/misc/SlidesCambridge.pdf
Structured deep models: deep learning on graphs and beyond

4. New Deep Learning Model

The original proposed deep learning model consisted of using the QA dataset which includes: timestamps, video segments, description of video segments, transcripts of the video segments, and questions/answers of the videos segments. The new training dataset will combine the QA dataset and the dependency relations between video segments. The hope is for the machine to learn about why the ingredients are used in order to create graphs that link concepts together.

I do propose a different way for the machine to learn via instructional cooking videos. I think that implementing a training dataset for learning to identify objects is the first step for the machine to learn reasoning because it needs to know what the object looks like. After it can correctly identify most any objects, I think another dataset should be able to show the properties of each object. For example, egg would be associated with certain concepts like "cook" → bake, fry, poach, etc. , "shell" → cracked or uncracked, and other words that entail other ideas about an egg.

I think making a sort of tree for each ingredient would be time consuming to create, but it would definitely teach a machine these small ideas about food. Along with this, they would be given images or videos showing actions common in cooking like a frying video, deep frying video, coating of ingredients video, and etc...This is a supervised learning technique which means the dataset consists of human labelled data. A non supervised approach would be an alternative if the proper dataset and concepts were selected, but the extracted features would have to be automatic (which I'm not sure how it's done).

This may all seem confusing, but this is the essence of the project I've discussed so far. The QA dataset alone produced a poor baseline result, we want to improve it. To do so we want to use the dependency dataset with the QA dataset so that the machine could have an improved prediction accuracy. This new model is still just a baseline model since we are simplifying the problem of video understanding to just comprehension of instructional videos. We want to be able to successfully generate graphs of these cooking videos because it serves as a precursor for video understanding as a whole (or so we hope).

5. YouCookWeb (web application)

YouCookWeb is a web application created for human data annotators to create dependency relationships between video segments. The dependency relationships serve as a baseline model and a ground truth. The baseline model is used to make a basic model, so that a more complex model can be created for a more accurate result. Ground truth are the parameters of the model, and the results of the training are checked against the real world data to improve accuracy. The dependency relations were chosen as elements to create a relational prediction network, steering away from a solely Question-Answering approach. The goal is for the machine to be able to link concepts and generate graphs with high accuracy.. The back-end framework involves using ASP.NET Core, Microsoft Entity Framework, and a SQL server.

` The creation of this web application serves as a data collection tool for annotators and then can be used as a predictive analytic tool for analysts. All the data annotations are saved into the database, and after all the data is collected it can then be part of the training dataset for the machine. Once the machine can pass baseline with an acceptable result then it can be validated and tested. From there we can then program all the data collected from the site to generate graphs that link together concepts the machine has learned from the datasets to visualize the data. At that point we could see what concepts and patterns the machine sees

in the data and can change around the algorithm's parameters to possibly improve it's predictions.

5.1. Prototyping YouCookWeb's Design

The idea of the design for the web application came after thinking how the tool can be displayed in the most intuitive way for the data annotator. What needs to be displayed on the screen for a data annotation? We need to know the video id, the video url, video segments, timestamps, descriptions of the video segments, and a way to link dependencies. We would ideally want the linked video segments be typed in pairs and want to show their relationship by placing an arrow with the two common video segments. For example if segment 1 is dependent on segment 2 then there would be arrow from 1 → 2. The video segments wouldn't just be text, but gifs from timestamp 1 to timestamp 2 in order for the data annotator to know which video shows what. Also, this way it is easier to spot errors because if you watch one segment and it has nothing to do with the next one then you know those are most likely not dependent on one another.

Multiple prototypes were drafted. We want this to be an interactive, yet informative web application. Not just a tool, but it also has information about the project for anyone to learn about. It will be a place to explore the datasets and anyone could contribute to the dataset since all of the data annotations should be getting reviewed before "accepting" it into the dataset. This is a rough idea (Figure 5) of how the layout could be and the way the actual dependency linking page could look. It will have a separate section to show the graphs of every video that was annotated to avoid confusing the data annotation page to the data visualization.

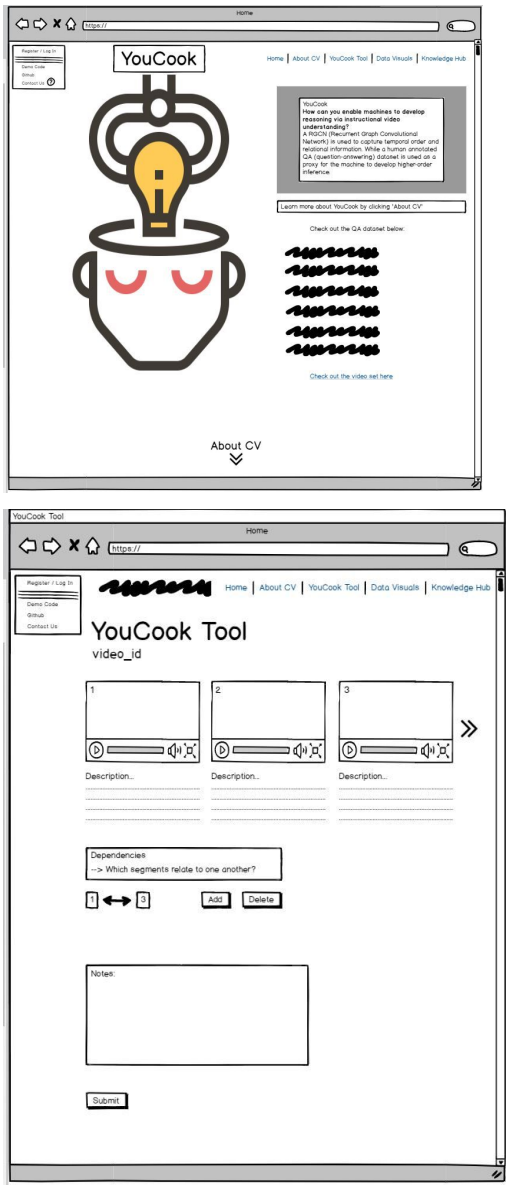
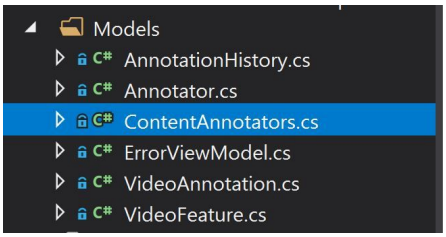


Figure 5: basic layout of the YouCookWeb web application

This web application also would not be just a platform to annotate data, but it would be multipurpose. We want the web application to be versatile for the data annotators themselves by making collections of data annotations (Figure 5.1) they've done so that they can be retrieved and annotations could be checked before use, Essentially, this is a system of handling data annotation information. More details about the features, functionality, and flow of the web application are in the next subsections.




```
namespace VideoData.Models
{
    public class ContentAnnotators
    {
        [Key]
        public int contentId { get; set; }

        public virtual IEnumerable<Annotator> Annotators { get; set; }
        public virtual IEnumerable<VideoFeature> VideoFeature { get; set; }
        public virtual IEnumerable<VideoAnnotation> VideoAnnotation { get; set; }
    }
}
```

Figure 5.1: models and collections of data for annotator, video annotations, and the whole videos with their associated information (id, segment, timestamp, etc.)

5.2. YouCookWeb Backend Development

When considering how to store and retrieve information there a few things to consider. I created an api that would be able to extract certain pieces of information from a database and post/save information to the database. The database will store information based on the schema (using Microsoft Entity Framework concepts) I created in the web application program. There are multiple classes that each serve a purpose in my code. For example, one class is an abstract class that another class can inherit information from to store additional but different data. Along with that, there's another class for the user "annotator", as well as, an audit class for all the data as collections (data annotations, video dataset, QA dataset, annotator accounts) which can be displayed on the homepage or a main page for anyone to sift through.

5.3. YouCookWeb Features and Flow

There are features we want the basic web application to have. The application should be able to make accounts for users. The only users of the site would be data annotators and then data checkers. The data annotators would be responsible for annotating the videos by finding dependency relationships between video segments. The data checkers are those checking through the videos to see if the data annotations are valid, check which annotators need to be paid, or are in charge of running the database itself.

The data annotators should be able to access all the data annotations they have submitted, how many are pending, how many are complete, which videos they submitted an annotation for, and they can see how much money they earned based on verified annotations. Those that are merely looking at the site can see the video dataset collection, video annotation verified collection, as well as, the eventual data visualization of the data annotations submitted. Those that are data checkers should be able to see which annotator has done which video, assign new videos for annotators, verify data annotations, add videos to the dataset, as well as, see which annotators need to be paid and so on.

The flow of the website is that there are multiple web pages explaining what Computer Vision is, the YouCook project, and then a data visualization page. The actual users are able to see the video annotation tool page, but anyone can see the video dataset. This is still the premature days of the application, but it is mostly to serve as a place to store a bunch of data information and an easy way to track who has done which video annotations, as well as, keeping tabs of who needs to be given compensation for their work. Making this online, accessible to anyone, can also just spread the word about this project and allow anyone to help annotate this dataset since more data makes for a diverse dataset.

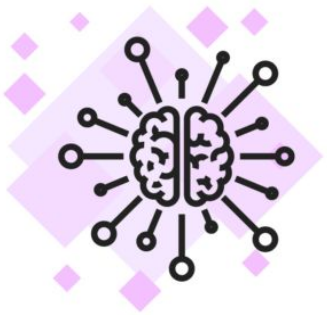
6. Conclusion

Computer Vision has been helpful in various object oriented applications such as object classification, object identification, object verification, object detection, and so on. It's also applied to things we may already use like facial recognition, augmented reality, image searching, handwriting recognition, and self-driving cars. Basically, its applications are numerous and it can be used within healthcare for medical imaging or security for surveillance to even a consumer in finding what shirt her favorite movie character is wearing. So far research has delved into all these mentioned topics, but mostly in the context of

still images. Videos are difficult because there is so much more information to work within a video of 30 seconds than in a still image.

Nonetheless, if a machine were able to understand videos it can improve these already researched topics and it can be a key to the puzzle of teaching machines to reason. If a machine could understand videos to an extent where it can identify an egg in a frame, what it is happening to it, why it is used, and other properties of an egg then it is truly comprehensive. When we see an egg we know it can be cooked in many ways, it has a hard shell that can crack, it has a unique texture, it has a unique savory taste, and it is used as a binder in many recipes. To a machine, it can identify an egg and what else? It doesn't know the context of it and what it can be. It just labels the object as "egg", but it's more complicated than that. As you can see, this task is no small feat, but if this project is successful then it can be a piece of the puzzle for video understanding.

YouCook has a way to go before it can be tested which is why it's hopeful creating this relation prediction network is the key to strengthen the dataset. The future for this project is that if the machine can learn from the instructional videos then it can surely learn from any video. The professor in charge could try to incorporate other video datasets to put the algorithm to the test, or expand this project to an all encompassing video understanding one. Either way this project is a useful one to understand how we could possibly enable a machine to reason by means of video data.



Note: These 2 sites have interesting information or an interactive nature for Computer Vision research.

[1]https://blog.e-kursy.it/deeplearning4j-cnn/video/html/#2.0

[2]http://web.eecs.umich.edu/~jjcorso/

Special Thanks:

-Kearns Center at the University of Rochester

-Xerox Engineering Research Fellows Program

-Chenliang Xu

-Tushar Kumar

-Cris Murray