Cooking for the Colorblind

Image Thresholding for Undercooked Ground Beef

Logan Jahnke[1] and Andrew Durden[2]

https://github.com/Adurden/Optical-Raw-Meat-Detector

[1] jahnke@uga.edu - Computer Science Department at the University of Georgia

[2] amd72531@uga.edu - Computer Science Department at the University of Georgia

Abstract

Colorblindness is a vision disability that affects approximately 1 in 12 men and 1 in 200 women around the world. While the most common type of colorblindness is not the stereotypical grayscale, those affected by it still have several daily challenges. One of those challenges is cooking ground beef. The redness of raw beef is indistinguishable with the brownness of cooked beef for those with deuteranomaly and protanomaly, or more commonly known as red-green colorblindness. This results in inconsistent doneness of beef sometimes producing undercooked meat and other times burnt meat. While burnt meat may not taste good, undercooked meat can cause food poisoning. Although there is abundant research into food quality assurance and safety, there is a noticeable gap in real time optical recognition of safely cooked ground meat. In recent years it has become more common for household chefs to have access to handheld recording devices such as smartphones with quality cameras and computing power capable of aiding in recognition of unsafe meat. Therefore, to empower those with colorblindness to cook ground meat and ensure safe and tasty meals, we utilize various computer vision techniques to determine the doneness of meat in real time.

Introduction

Foods with a high protein content provide an ideal environment for the multiplication of pathogens. In the United States, foods with high risk for food poisoning include raw shellfish, raw eggs, and rare meats[3]. It is estimated that over 48 million people get food poisoning every year, of those 126,000 are hospitalized and 3,000 die[4]. It is also estimated that at least 20% of food poisoning is from home cooking; however, it is suggested that this number may be higher due to larger reporting rates for food poisoning caused by restaurants[5]. This means that many people are accidentally poisoning themselves, whether it be from a bad batch of food or from poor preparation.

There are several ways to protect yourself against food poisoning including washing your hands prior to food contact, keeping raw foods separate from prepared foods, and many more. One of the most important prevention methods is cooking food to a safe temperature, especially rare meats. The best way you can kill harmful organisms in most foods is by cooking them to the right temperature which can be determined with a food thermometer. While the majority of Americans own a food thermometer, as little as 10% of Americans use their food thermometer on a regular basis[6]. Therefore, we hypothesize that the most common way of telling if meat is done at home is by simply examining the color of the interior and exterior of the meat. In ground beef, our focus, the exterior suffices while in other meats such as chicken breast, the interior should be examined.

---

[3] Henze, 2013.

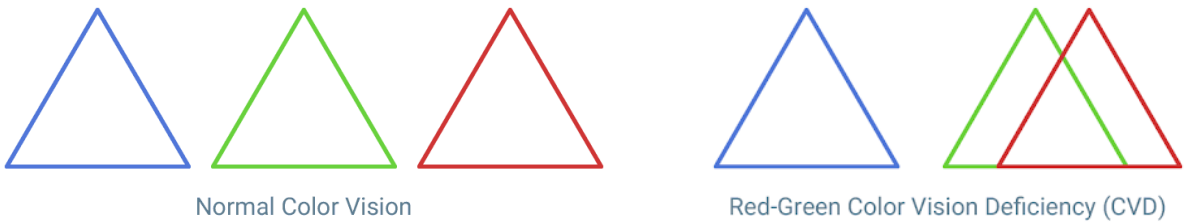
[4] Centers for Disease Control and Preventions, 2018.

[5] Preidt, 2014.

[6] Lando et al., 2016.

While it may be a trivial task for most people to check for redness on the exterior of ground beef while cooking, it can be a difficult task for those with color deficiencies. Normal color vision is defined as trichromacy and uses all three types of light cones correctly. People with normal color vision are known as trichromats.

Color deficiency is almost always a form of dichromacy which is when only two of three types of light cones work correctly. People with dichromatic color vision are known as dichromats. The lack of ability to see a color is the easiest way to explain dichromacy, but in reality, dichromats have an entire spectrum of color that is either inhibited or disabled. Similar to an image intensity, these spectrums are split into three colors: red, green, and blue. There are three types of dichromacy, each struggling with a different spectrum. The different anomalous conditions are protanomaly, which is a reduced sensitivity to red light, deuteranomaly which is a reduced sensitivity to green light and is the most common form of colorblindness and tritanomaly which is a reduced sensitivity to blue light and is extremely rare. Those with the most common forms of colorblindness, protanomaly and deuteranomaly, which are conditions that inhibit a spectrum rather than disable a spectrum, tend to see the world in a similar way. The red and green spectrums overlap in those with protanomaly and deuteranomaly which is why these two forms of color deficiency are collectively known as red-green colorblindness. People with a red-green colorblindness see the world in murky greens while blues and yellows stand out. They easily confuse greens, reds, browns and oranges. They can also easily confuse blue and purple[7].



Visual representation of the three light cones in someone with trichromacy and someone with a form of red-green colorblindness. Photo credit: Enchroma.

Therefore, it is easy to see that those with protanomaly and deuteranomaly likely have trouble determining raw ground beef from cooked ground beef. The redness of the rare beef can be often overlooked when in a grouping of browned beef. Although there are no official statistics on food poisoning for people who are colorblind, because of the commonality in checking the color for doneness of ground beef, it is likely that those who are colorblind are more likely to undercook their beef and as a result, get food poisoning.

[7] Colour Blind Awareness.

While the most common personal method for checking doneness of meat is through vision, the food industry has been attempting to automate the measure of meat quality using computer vision technology for several years. With the advantages of rapid, accurate, non-contact and non-destructive analysis of food, there have been many studies into different computer vision techniques. It also provides a high level of flexibility but low cost solution to ensuring food is cooked and stored properly.

Left image: how a person with normal vision sees ground beef during an intermediate cooking stage.
Right image: the same image passed through a dichromacy filter. This is how someone with red-green colorblindness sees the same ground beef during an intermediate cooking stage.

Unfortunately, there is no easy access to an autonomous system for personal use. Therefore, we compare and implement several computer vision techniques to determine the doneness of ground beef with plans to eventually create a real-time phone application for colorblind users.

Problem and Approach

Our problem is twofold: we aim to autonomously determine where ground beef is within an image, and we then quantify the portion of the meat which is uncooked. Of these two goals succeeding in the first should improve the latter by allowing us to ignore background pixels which could be misclassified by color as uncooked meat.

In order to discover which computer vision method will provide rapid and accurate results, we first need to split the cooking process into a discrete experiment. In order to be as realistic as possible, we took a video of the cooking process in suboptimal lighting and at a side angle starting with raw ground beef and finishing with overcooked beef. Then, we split the video into single frames for every 30 second time interval. Furthermore, we also tested beef using the same

process with optimal lighting and a near-birds-eye view for the camera. Therefore, we have two sets of images, one with suboptimal lighting with the beef at an angle (Suboptimal Set), and the other with optimal lighting with 100% of the beef in frame (Optimal Set). Some possible obstacles that may influence disparities among a set of images include moving the fry pan due to stirring the beef and the smoke from the cooking process; however, neither of these should affect the final results. In order to ensure that the smoke does not affect any results, we included a fan to dispel the smoke from the camera in the Optimal Set of images, but not in the Suboptimal Set.

Unfortunately, the Suboptimal Set used ground venison (deer), and while achieving fairly good results, it is not a good indicator of whether our algorithms can determine cooked ground beef from uncooked ground beef. Therefore, the following experiment will only be conducted on the Optimal Set.

In order to score any computer vision method we implement, we first hand segment the uncooked regions and cooked regions for the set of images. Each image is given "browning ratio" where the higher the percentage, the more brown the meat, meaning more cooked. The browning ratio will be computed by dividing the total number of cooked meat pixels by the total number of meat pixels. The background pixels, such as the fry pan and the stove top, should be ignored.

$$\text{Browning ratio: } BR(i) = \frac{C}{C+U} \mid C = \text{cooked meat and } U = \text{uncooked meat.}$$

With the frames hand segmented, an intersection over union (IoU) of the hand segmented frames and the algorithm segmented frames will tell us how good our algorithms are at segmenting all meat from the non-meat in a frame. To compute, find the number of pixels in the intersection of the manual and the autonomous segmentations, and divide it by the number of pixels in the union of the manual and the autonomous segmentations. Therefore, an IoU score of 1.0 is a perfect segmentation where all pixels in the autonomously segmented image are exactly the same as those in the corresponding manually segmented image, and an IoU score 0.0 is a segmentation where the pixels in the autonomously segmented image never overlap with those in the corresponding manually segmented image.

$$\text{Intersection over Union: } IoU = \frac{I}{U} \mid I = \text{pixel count in Intersection and } U = \text{pixel count in Union.}$$

Additionally, someone with red-green colorblindness hand segments the uncooked regions from the cooked regions for the set of images. Just like before, each image is given a "browning ratio". This set of hand segmentations gives us the typical error rate for those who have red-green colorblindness.

Finally, the score of our algorithms will be based on three numbers: The average error, the average improvement, and the average improvement during frames in which both uncooked and cooked meat are present.

$$\text{Average Error: } AE_x(i) = |BR_{Hand\ Segmented}(i) - BR_{x\ Segmented}(i)|$$

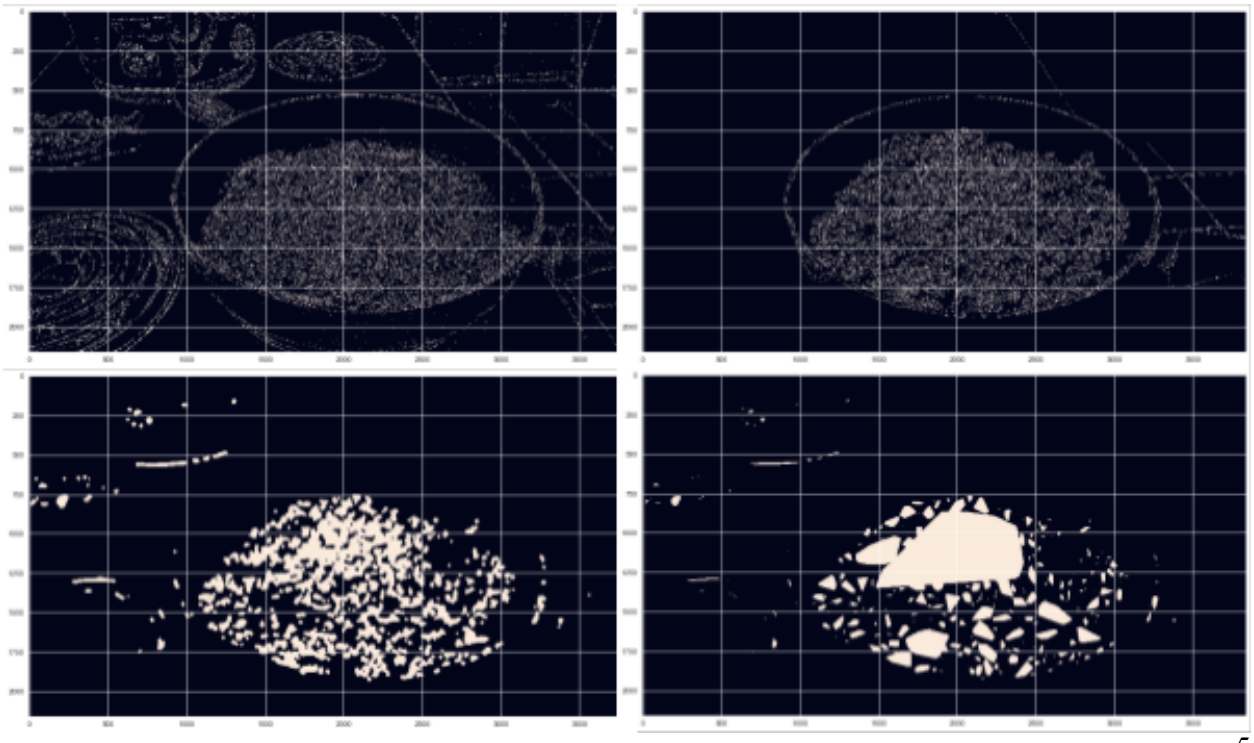
$$\text{Average Improvement: } AI = \frac{\sum_{i=1}^{20} |AE_{Colorblind}(i) - AE_{Algorithm}(i)|}{20}$$

$$\text{Average Improvement in conflicted frames: } AI_c = \frac{\sum_{i=8}^{12} |AE_{Colorblind}(i) - AE_{Algorithm}(i)|}{5}$$

Determining What is Meat

Our first goal was to determine what is meat. For this we tried a few approaches. The first approach we tried was an edge map dilation technique to create a generous mask. This was often too sensitive to parameter selection and very fickle in terms of yielded mask accuracy. We then transitioned our efforts towards histogram based object detection techniques in several color encodings.

The goal of edge map dilation was to use the prior knowledge of the texture of ground meat to properly localize the meat in the image. Our process was to create an edge map using an algorithm such as the Canny Edge Detection with a low value of sigma. The edge map should then have a large mass of noisy edges in the area of the image containing meat, while most of the rest of the image would have fewer noisy edges.



The edge map (top left), erosion (top right), dilation (bottom left), and convex hull (bottom right) steps of the edge map dilation segmentation process applied to the first frame of a video.

From here we applied morphological operations like dilation and erosion to try to remove extra edges in other regions of the image while keeping the meat as one solid region. Finally we'd apply a convex hull to the objects within the image in hopes of retrieving a mask. Visibly, this process was inaccurate and required frame specific manual tuning. Therefore we shifted our focus to try to find a more accurate and robust technique.

To try to find appropriate thresholds we looked at the distribution of values in each RGB color channel in hopes to see where the meat pixels exist in the color distributions. When we look at each channel separately for uncooked meat, image background, and cooked meat we see very defined distributions for the meats but no gap in the image background. This implies an overlap which would cause a large amount of noise in a segmentation on any of the channels alone.

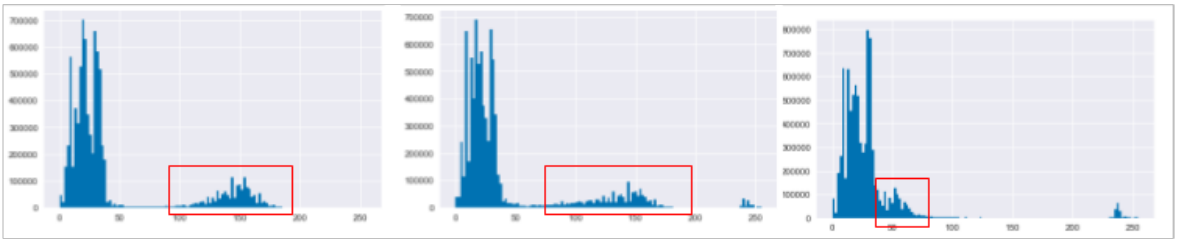
These are channel distributions for the Red, Green, and Blue channels for the uncooked region (top) background (mid) and cooked region (bottom).

We then looked at it as a two dimensional distribution of the red and green channel (as the blue and green channels are similar within the meat portions). What we see is a very isolated region while the meat is raw, but as the meat cooks it blends in more and more with the background values in RGB encoding. We followed the same analysis process in HSV encoding hoping to find more defined threshold options in the three dimensions. However the channels still overlapped

between the object and background areas which led to difficult to determine thresholds in three dimensions in both encodings.

```
RED_LESS_GREEN(image)
   // Separate the channels
   channels = get_channels(image)

   // Subtract green channel from red channel
   for all pixels (x,y) in image
      channels(red)(x,y) -= channels(green)(x,y)

   // Create histogram from modified channel
   histogram = calc_histogram(channels(red))

   // Find all local mins
   mins = []
   for prev, current, next in histogram
      if prev > current < next
         mins.add(current)

   // Find closest local min to 40 and 200
   min_difference = INT_MAX
   max_difference = INT_MAX
   min_threshold = 0
   max_threshold = 0
   for min in mins
      if |min - 40| < min_difference
         min_difference = |min - 40|
         min_threshold = min
      if |min - 200| < max_difference
         max_difference = |min - 200|
         max_threshold = min

   // Threshold for all meat
   all_meat = thrshld(channels(red), min_threshold)

   // CCL to remove noise
   components = ccl(all_meat)

   // Remove small objects to isolate meat
   all_meat = components.filter(min = 1/5 of largest)

   return all_meat
```

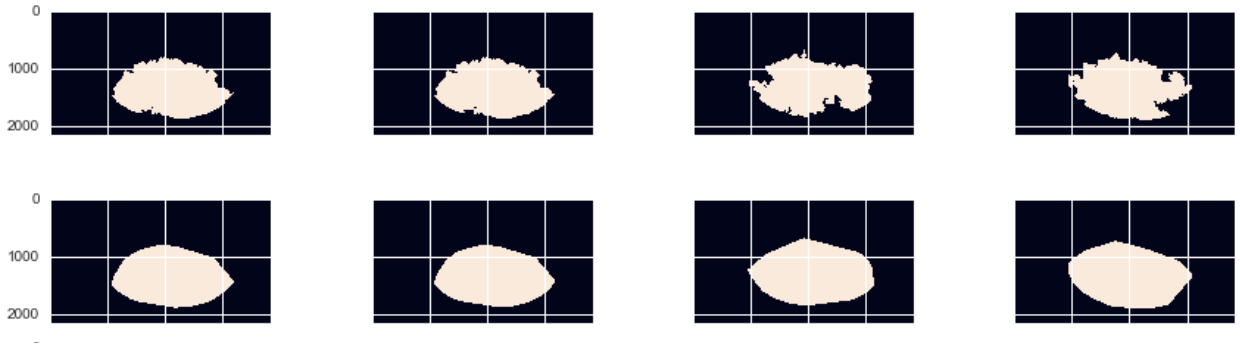
Our solution was a novel re-encoding process which we have dubbed ‘red less green’ encoding. The encoding process creates a grayscale image where the value is the difference between the red and green channels. It is an effort to preserve uniqueness of color values in the meat regions, while reducing the dimensionality from three to one allowing us to only need upper and lower thresholds for a single channel. We first separate the RGB channels from the original image. Then we subtract the green channel from the red channel and throw out the blue channel. Therefore, we are left with one channel that we call red less green. It ensures that pixels with inherently high red values that are not necessarily red are removed from our thresholds. For example, a white pixel would have the RGB value (255, 255, 255). A white pixel is not red so we would like to filter it out before we do thresholding. So we isolate the red channel and then subtract the green channel, and we are left with an intensity value of 0, or a completely black pixel. When the distribution in this new channel was viewed there is a very clear mode corresponding to the meat in the image, and while it shifts over time as the meat cooks the mode still remains mostly separate from the background.



Above is the distribution of values in the Red Less Green encoding as meat cooks with the mode representing the meat shown in the red box.

After manually viewing the distribution over all of our data we seeded our algorithm with a lower and upper bounds of 40 and 200 respectively. Our algorithm then looked for local minima in the distributions of values nearest the seeds to use as threshold values in each frame. This gives us a slightly under segmented image which includes a small amount of background noise.

To combat this noise we used a method of combined connected component labeling and object size filtering to create masks. This process was generally accurate, save for frames occluded by the spatula or where the meat in the pan is broken into separate areas. To improve our accuracies on these frames we used a convex hull to create single contiguous masks.



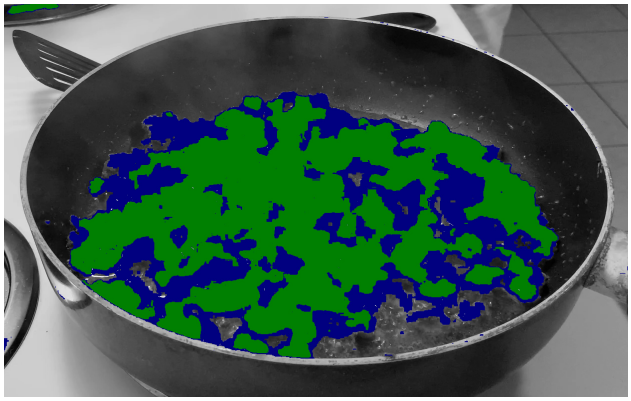
The final masks (top) and their convex hulls (bottom) for the first four frames of our ideal video.

We used the IoU scoring method to judge the accuracy of our masks. These figures can be found in the results section. With this algorithm finalized we could filter the area of the image we considered for the following cookedness detection algorithms.

Determining What is Uncooked Meat

The second part of goal is to quantify the portion of the meat which is uncooked. To do this, we will use several different algorithms from the literature. In some scenarios, a simple image segmentation is sufficient for telling if food is thoroughly cooked. Mogol analyzes potato chips browning ratio similar to our ground beef browning ratio. Because of similar scoring methods, Mogol's image segmentation method may be a simple and ideal choice[8].

Mogol's image segmentation first converts the frame from RGB (red, green blue) color space, which is on a scale from 0 to 255 for each value, to HSV (hue, saturation, value/ brightness) color space, which is on a scale of 0 to 360 for hue, and 0 to 100 in saturation and brightness. It then thresholds the entire image for uncooked meat in a specific color range, then it thresholds the entire image for cooked meat in another color range. Finally, it merges these two binary images back into the original image to display an overlay of the results on top of the original image in



An example of the resulting image for the channel re-encoding algorithm. Mogol's algorithm produces a similar result.

[8] Mogol, 2013.

```
MOGOL(image)
   // Convert from RGB to HSV
   hsv = convert(image, RGB2HSV)

   // Determine ranges for uncooked meat
   l_min = H:0, S:25, V:25
   l_max = H:4, S:100, V:100
   u_min = H:346, S:25, V:25
   u_max = H:360, S:100, V:100

   // Threshold the color range for uncooked meat
   luncooked = thrshld(hsv, l_min, l_max)
   uuncooked = thrshld(hsv, u_min, u_max)

   // Merge thresholds into uncooked meat image
   uncooked = merge(l_uncooked, u_uncooked)

   // Determine ranges for cooked meat
   c_min = H:10, S:35, V:39
   c_max = H:24, S:100, V:70

   // Threshold in the color range for cooked meat
   cooked = thrshld(hsv, c_min, c_range_max)

   // Overlay onto original image
   for all pixels (x,y) in image
      if uncooked(x,y) == 1
         image(x,y) = green
         uncooked_pixels += 1
      else if cooked(x,y) == 1
         image(x,y) = blue
         cooked_pixels += 1
      else
         image(x,y) = convert(image(x,y), RGB2GRAY)

   return image
```

```
DU(image)
   // Set average to zero
   avg_red = 0
   total_pixels = 0

   // Segment for meat
   segmented = RED_LESS_GREEN(image)

   // Overlay onto original image
   for all pixels (x,y) in segmented
      if segmented(x,y) is not black
         avg_red += red_channel(segmented)(x,y)
         total_pixels += 1

   // Divide to find average
   avg_red = avg_red / total_pixels

   return avg_red

QUADRATIC_REGRESSION(red_value)
   // Get A, B, and C for regression
   A = 10.277144
   B = -0.09767746
   C = 0.0002294

   // Compute y
   y = A + B * red_value + C * (red_value^2)

   return y
```

grayscale. This is so people who are colorblind can easily tell which portions of the meat and raw and which portions of the meat are cooked. During the overlay, the number cooked pixels and uncooked pixels are counted to determine the browning ratio.

An even simpler algorithm that may prove effective is Du's mean color measurement. Rather than spotting specific areas of redness, Du simply takes the mean and standard deviation of color over the entire image. For more accurate measurements, Du took the mean and standard deviation of several small sub-images of the total image in order to check for undercooked food. This is similar to adaptive thresholding[9].

Rather than taking the mean redness over the entire image, we take the mean redness over the segmented meat portions of the image as the background pixels could significantly impact the redness of the image. Du's mean redness algorithm takes the red channel's intensity of each pixel in the segmented meat and averages them together. This average value is then fit into a quadratic regression to determine the browning ratio.

In addition to Mogol's and Du's algorithms found in literature, after some experimentation we created our own algorithm for determining the browning ratio of meat using our red less green encoding. Using a similar technique and prior knowledge we set another minimum threshold to segment the undercooked meat away from the cooked meat. Finally, just like in Mogol's

[9] Du, 2016.

algorithm, we combine the two resulting images as an overlay on top of a grayscale version of the original image. We also count the total uncooked pixels and total cooked pixels in order to compute a browning ratio.

Results

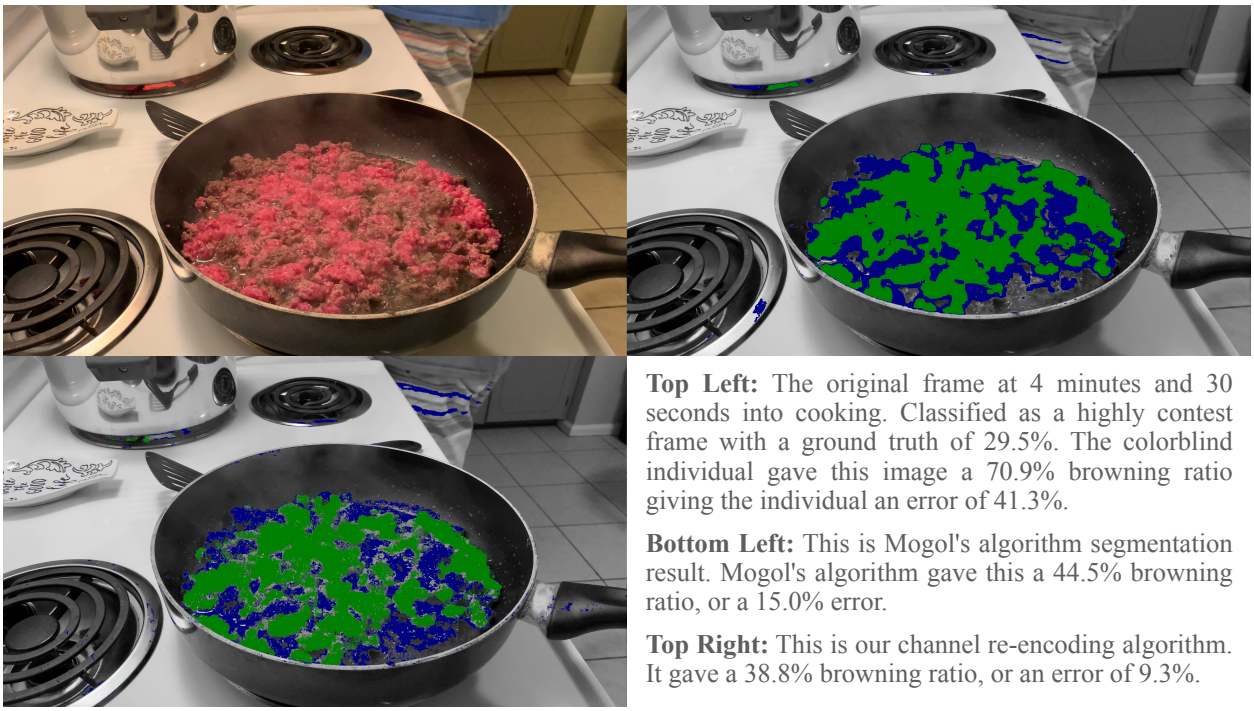
Our meat finding algorithm was scored based purely on their IoU with the ground truth manual segmentations. In general they were very accurate with an average IoU of .81. However in many images the IoU drops due to occlusion caused by a stirring instrument, or due to the meat existing in multiple disconnected areas. Our attempt to fix these inaccuracies using convex hulls raised our average IoU to .84.

Frame	IoU	IoU-CH
0	0.96	0.94
900	0.96	0.94
1800	0.89	0.84
2700	0.91	0.88
3600	0.92	0.87
4500	0.86	0.78
5400	0.91	0.88
6300	0.88	0.89
7200	0.89	0.90
8100	0.92	0.95

Frame	IoU	IoU-CH
9000	0.92	0.81
9900	0.88	0.95
10800	0.56	0.66
11700	0.85	0.91
12600	0.39	0.51
13500	0.87	0.95
14400	0.51	0.64
15300	0.89	0.94
16200	0.86	0.88
17100	0.47	0.79

Before mentioning the performance of our cookedness algorithms, it is important to note our baseline. Recalling our scoring method, ideally, we would have a low error rate against the ground truth and a high improvement rate over the typical colorblind error. The overall error rate for a typical red-green colorblind individual was 6.7%, and the overall error rate for that same individual in frames that were highly conflicted, containing a high mix of cooked and uncooked beef, was a scary 17.0%. Furthermore, during the first highly conflicted frame, the colorblind individual greatly overestimates the ratio of cooked to uncooked beef by 41.3% thinking that the meat is around 71% cooked when in reality it just 29.5% cooked.

Overall, all three algorithms did something better than our typical colorblind error. However, some algorithms did far better than others. Mogol's algorithm was excellent at grabbing cooked meat pixels when there was no uncooked meat pixels in frame. The flat HSV thresholding was extremely effective at the lower and higher ends of the spectrum (highly populated with uncooked meat or highly populated with cooked meat). It was also fairly good at thresholding in



Top Left: The original frame at 4 minutes and 30 seconds into cooking. Classified as a highly contest frame with a ground truth of 29.5%. The colorblind individual gave this image a 70.9% browning ratio giving the individual an error of 41.3%.

Bottom Left: This is Mogol's algorithm segmentation result. Mogol's algorithm gave this a 44.5% browning ratio, or a 15.0% error.

Top Right: This is our channel re-encoding algorithm. It gave a 38.8% browning ratio, or an error of 9.3%.

conflicted frames with a typical improvement rating of 5.5%. Overall, Mogol's algorithm showed great improvement over a typical colorblind error of 2.4% and maintained a low error rate of 4.4%. However, we are unsure if these results would remain accurate under a different lighting setup.

Du's algorithm improved on the typical colorblind error in frames that were conflicted with high mix of both uncooked and cooked meat, but was highly inaccurate when there was clearly totally uncooked meat or totally cooked meat. The average error for Du's algorithm was an abysmal 9.8% with an overall average improvement of -3.1%. That's right, Du's algorithm was less accurate than that of a colorblind person. The only bright side to Du's was the average improvement rate during conflicted frames which was 4.1%. Still lower than Mogol's, but at least it is better than a colorblind person. Du's algorithm may perform better if given more training data rather than just one set of images.

Our red less green algorithm was highly accurate in all aspects of scoring. However, towards the later frames, our algorithm would leave out some cooked pixels. While this did not affect the scoring as there were no uncooked pixels in the frame, our browning ratio was matching the ground truth, but it did not accurately grab all of the meat in frame. This is likely because as the meat becomes more and more cooked, the redness in our new encoded channels begins to blend in with background pixels making it harder to threshold based on a histogram analysis. Luckily, all the meat is already done so the browning ratio score is still a perfect match. Overall, our channel re-encoding algorithm had a low average error rate of 3.5%, almost an entire percent

better than Mogol's algorithm. It also had an overall average improvement rate of 3.3% and a high 10.4% average improvement rate in conflicted frames.

Algorithm	Mogol's	Du's	Channel Re-encoding
Avg. Error	4.4%	9.8%	3.5%
Avg. Improvement	2.4%	-3.1%	3.3%
Avg. Improvement in conflict frames	5.5%	4.1%	10.4%

Overall, Mogol's and our channel re-encoding algorithm performed well enough to aid a colorblind individual. However, Du's algorithm at times is detrimental to colorblind individuals in that it is actually less accurate than them.

Conclusion and Future Works

Using computer vision to aid colorblind cooking is a realistic endeavor. With results already being positive and improving upon the error of a colorblind individual, with a few more modifications, a real-time mobile application is not far-fetched and can be created to provide live feedback to its user.

To ensure the accuracy of these algorithms, more testing data should be created, and all data should be hand segmented by a group of people rather than just one person to remove biases and ensure accurate baselines. Furthermore, fully combining our high IoU meat finding algorithm with the cookedness algorithms will improve their focus by eliminating noise. Finally, using histogram equalization in our high IoU algorithm may provide more accurate local minimums for thresholding. With encouraging results, an aid for colorblind cookers is close to reality, and, with a few more adjustments, a real-time mobile application can be created to help those with color deficiencies everywhere.

Citation of References

Du, C.-J., et al. “Quality Measurement of Cooked Meats.” Computer Vision Technology for Food Quality Evaluation, 2016, pp. 195–212., doi:10.1016/b978-0-12-802232-0.00008-6.

“Food Safety.” Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 16 Feb. 2018, www.cdc.gov/foodsafety/foodborne-germs.html.

Henze, Martha M. ..MS, RD, and Maria Pacheco PhD. “Food Poisoning.” Magill’s Medical Guide (Online Edition), 2013. EBSCOhost, proxy-remote.galib.uga.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=ers&AN=89093412&site=eds-live.

Lando, Amy, et al. “2016 Food Safety Survey” Food and Drug Administration, 2016, https://www.fda.gov/downloads/Food/FoodScienceResearch/ConsumerBehaviorResearch/UCM529481.pdf.

Mogol, Burçe Ataç, and Vural Gökmen. “Computer Vision-Based Analysis of Foods: A Non-Destructive Colour Measurement Tool to Monitor Quality and Safety.” Journal of the Science of Food and Agriculture, vol. 94, no. 7, 2013, pp. 1259–1263., doi:10.1002/jsfa.6500.

Preidt, Robert. “Restaurants Pose Double the Risk of Food Poisoning?” WebMD, WebMD, 9 Apr. 2014, www.webmd.com/food-recipes/food-poisoning/news/20140409/restaurants-pose-double-the-risk-of-food-poisoning-compared-to-homes--study.

“Types of Colour Blindness.” Colour Blind Awareness, www.colourblindawareness.org/colour-blindness/types-of-colour-blindness/.

Division of Work

Everything in the "Determining What is Meat" section including code (Python) that deals with the section was done by Andrew. Everything in the "Determining What is Uncooked Meat" section including code (OpenCVSwift) that deals with the section was done by Logan. The project report and presentation slides were written together by both Andrew and Logan.