Combating Obesity Through the Use of Machine Learning Calorie Detecting Algorithms

Adrian Sujkovic

Department of Computer Science, Binghamton University

CS 301: Ethical, Social, and Global Issues in Computing

Dr. George Weinschenk

May 11th, 2022

Abstract

Getting Americans to understand the number of calories they consume on a daily basis can greatly help reverse the current obesity pandemic as many don't have the knowledge of the nutritional content of most foods. Using convolutional neural networks holds promise as the technology can allow the average consumer to calculate the number of calories they consume by simply taking a picture of their food. This technology is possible thanks to researchers in the field such as Takumi Ege, Keiji Yanai, and Kaimu Okamoto who have studied CNN technology in relation to estimating calories in food by using datasets of pictures of various dishes. Two algorithms these researchers used are YOLOv2 and DeepLabv3+ which are compared in their effectiveness at estimating calories in dishes. The YOLOv2 algorithm beats the DeepLab V3+ in terms of estimating the calories in a given dish of food because it can detect more complex meals quicker and more accurately, allowing users to better track their nutrition and take care of health problems. This comes down to YOLOv2's efficiency in processing input quickly. Given the relatively new and rapidly evolving nature of the field of deep learning, these algorithms are both still far from perfect, therefore more research is still needed until the technology can truly be adopted on a large scale.

Combating Obesity Through the Use of Machine Learning Calorie Detecting Algorithms

The prevalence of obesity in American adults amounts to over 40%, entailing a BMI of over 30 (Hales, 2020). Some massive contributors to this statistic include the nation's overconsumption of fast food and its lack of exercise. Even when attempting to diet, people often lack the knowledge of the caloric content of food, which results in underestimation of calories and overeating. When people fail to lose weight despite eating less, they become demotivated and keep eating more. Machine learning can help combat this issue using algorithms to estimate calories on a plate. Two examples of such algorithms include You Only Look Once (YOLO) v2 and DeepLab V3+, which identify objects in images using deep learning algorithms that learn from thousands of images like a human brain. The YOLOv2 algorithm beats the DeepLab V3+ in terms of estimating the calories in a given dish of food because it can detect more complex meals quicker and more accurately, allowing users to better track their nutrition and take care of health problems. The workings of both algorithms take an image as an input and return the name of an object as an output. The difference, however, comes in the approach taken by each algorithm in order to achieve the desired output.

Alternative Technology

DeepLab V3+, a deep learning algorithm, uses an input image to return semantic labels (classes) for all objects in said image. Developers then use semantic labels to compute the caloric content of food in an image. Deep learning, a type of machine learning, uses substantial amounts of a specific input to mimic the neurons of a human brain. The algorithm starts with a blank slate, but after enough input, it builds up enough "neurons" and connections to tackle problems given to it. (Farsal et al., 2018, p. 1). For example, if a human studies dishes of food and their caloric content, they then can memorize the nutritional content of a dinner plate. Deep learning

algorithms such as DeepLab V3+ similarly receive an input of thousands of food images, allowing them to predict the caloric content of the meals within said images without any involvement from a user. The extent of the deep learning technology goes much further than just inputs and outputs.

Now that we know how deep learning works, we can delve deeper into the specific type of neural network used by DeepLab and YOLO. DeepLab V3+ uses Convolutional Neural Networks (CNNs) to process information, which have shortened their training time from weeks to merely hours in recent years (Farsal et al., 2018, p. 2). What makes these neural networks so powerful lies in their ability to translate topological inputs into algorithm-friendly code, turning 2D images into 3D objects. A CNN contains four key parts: a convolution layer, a pooling layer a fully connected layer, and a loss layer.

The bulk of the computation resides within the convolution layer where the neural network applies a filter onto an input image, extracting certain features from the image depending on the type of algorithm.

Figure 1

Process of Convolution in a CNN

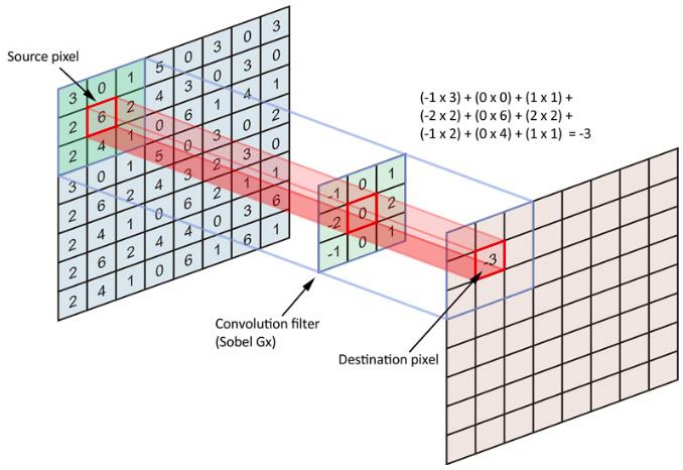
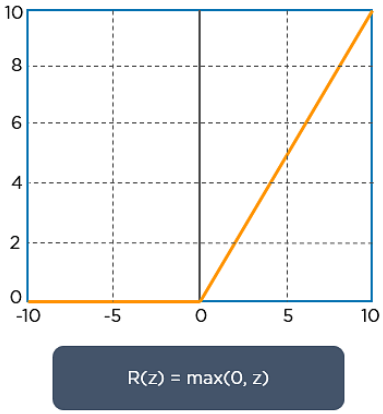


Figure 1 (freeCodeCamp, 2018) shows this filtering process, called convolution. The algorithm multiples a filter, or matrix with a specific layout of numbers, with a block of pixels in the source image to find the element-wise cross product. The algorithm stores the result in the output panel. The process then repeats for the rest of the reachable pixels in the input image. In the case of a non-monochrome image, or an RGB image, the process conducts on three separate inputs: a red channel, a green channel, and a blue channel. In the CNN, this process repeats several times, whether for down sampling or up sampling the image.

Figure 2

ReLu (Rectified Linear Unit) Graph



In the convolutional layer, the algorithm down samples the image. After down sampling, the resulting output panel goes through a ReLu, or rectified linear unit function, as shown in Figure 2 (Biswal, 2022). This function simply turns any negative pixels to zero and preserves positive pixels. Non-linearity allows CNNs to operate smarter and freely without the bounds of a straight line.

The pooling layer further simplifies the input into a smaller piece of information, just like the convolutional layer. Next the fully connected layer upscales this reduced reading of the original image. The loss layer then calculates the error rate of the actual result compared to the

expected result. These CNNs make image processing quick and accurate, despite the limitation of a 2D camera attempting to analyze a 3D plate of food (Arora et al., 2020, p. 750).

Figure 3

Convolution Formula

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

The formula shown in Figure 3 (Khandelwal, 2018) summarizes the process of convolution. The two summations signify the rows and columns of the filter matrix, usually with a lower bound of one and an upper bound of the size of the matrix. In the example of a three-by-three filter, the summation will have a lower bound of one and an upper bound of three on both axes. The formula takes the m by n entry of the weight, or filter, multiplied by the corresponding pixel in the input image. The formula then takes the summation of all nine entries and then repeats the process for the entire image. When we refer to tweaking the weight in backpropagation, we refer to the entries in this filter matrix. Some formulas include a variable called a bias that the formula adds after multiplying weight and input at a pixel. This bias offers another form of tweaking the algorithm to perfect the output that covers more than just one pixel.

CNNs make up the bulk of the method used by Okamoto et al. to estimate calories in a dish of food. Kaimu Okamoto (2021), a resident of Tokyo, Japan, performs research at the prestigious University of Electro-Communications in Tokyo. Okamoto has written a couple of papers within the last eight years on using deep learning to analyze food related trends. Keiji Yanai, a professor of Okamoto at the University of Electro-Communications in Tokyo, is a leading researcher at the university. Yanai's first paper in the IEEE database dates back to 1998, and he has published 60 papers since then. Okamoto and Yanai's paper on region-based food

calorie estimation for multiple-dish meals uses the DeepLab v3+ region segmentation model (Okamoto et al., pp. 19).

Keiji Yanai (2021), who maintains a lab named after him at the University of Electro-Communications in Tokyo, authored a similar paper with a student by the name of Takumi Ege. Ege, another student of Yanai at his university, focuses on tackling similar problems by using CNNs to estimate calories in foods. The only difference comes in Ege residing in Chofu, Japan. Ege and Yanai's paper on multi-task learning of dish detection and calorie estimation deploys the YOLOv2 CNN-based object detector (Ege & Yanai, pp. 2-3). The similarity in the authors of these two papers on DeepLab and YOLO, along with their similar application of the networks, helps to better compare the two algorithms.

Support

To fully grasp the difference between YOLOv2 and DeepLab v3+, we must first understand why they were chosen for the task of image processing. Both algorithms use deep learning. Deep learning. the computing field of choice for image recognition, tops other fields of artificial intelligence due to its ease of use for consumers. The key difference between machine learning and deep learning comes in the form of error detection. If something goes wrong in a machine learning algorithm, the user must make changes to the algorithm. In deep learning, the algorithm uses an algorithm to fix and learn from its own mistakes, making the algorithm more user friendly. This allows the algorithm to train much quicker and therefore show up in applications quicker.

The goal of a CNN is to mimic a human brain. The network contains nodes with connections, mimicking neurons and their synapses. Researcher Fei-Fei Li (2015) explains this topic in depth in her ted talk on CNNs. Born in Beijing China, Li moved to the United States at

15. She majored in physics at Princeton University and funded her studies by working in her parents' dry-cleaning store on weekends. Li went on to receive a PhD from the California Institute of Technology and is now a leading figure in the field of AI. In her talk on CNNs, Li showed that her team of researchers created an input size of 1.5 million photos to show the network to mimic the number of things seen by a baby as it grows. She claims that the current capability of the CNN is that of a three-year-old, and that the algorithm can only get stronger from there (Li).

The only drawback in the power of CNNs comes in their difficulty to understand. Ted Lewis and Peter Denning (2017) sum the mechanics up succinctly. Lewis received a BS in mathematics, followed by a PhD in computer science. After an extensive career in software, Lewis now teaches at the Naval Postgraduate School and writes for the IEEE. Born in Queens, New York, Denning received his bachelor's in electrical engineering at Manhattan college. He then went on to pursue a PhD in electrical at engineering at the prestigious MIT. Denning has since written 340 technical research papers. In Lewis and Denning's Communications of the ACM article on learning machine learning, they claim that you do not program a neural network: you teach it how to learn. These CNN algorithms require a massive amount of input and a long time to train, but once they have been properly taught, they run very quickly (Lewis et al., pp. 25). The inner workings of these networks involve many steps.

Figure 4

Backpropagation in a Neural Network

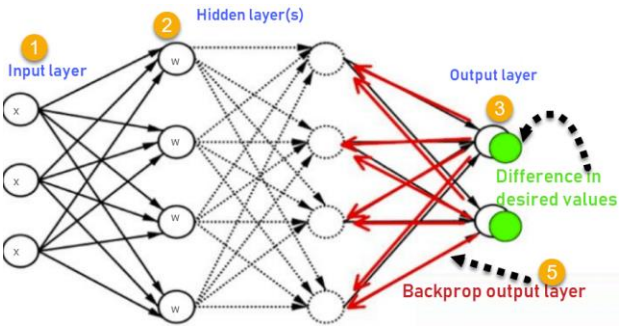


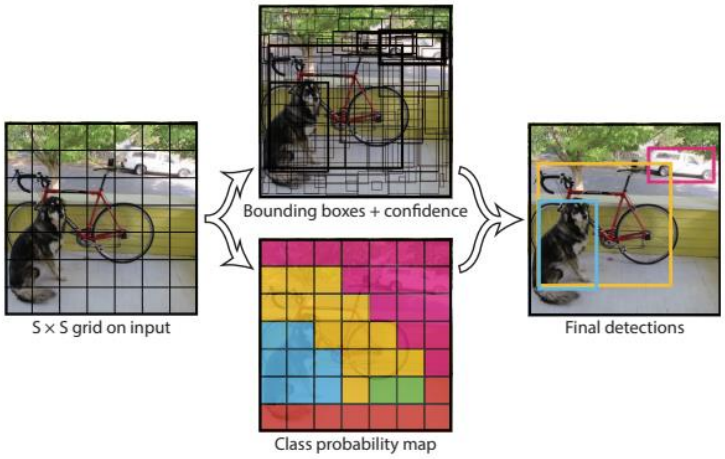
Figure 4 (Johnson, 2020) shows a typical neural network, where the input starts on the left side, traveling through neurons in the network until the network reaches an output. The network then compares its actual output to its expected output, then traverses backwards through the layers, adjusting weights such that the error rate from the previous iteration decreases as much as possible. This backwards traversal, or backpropagation, looks for the biggest weights in the network. The more impactful a weight seems, the easier for the algorithm to change that one weight rather than multiple other smaller weights. If a researcher gives the algorithm a well-organized dataset with correctly labeled expected outputs, making these tweaks to the weights allows the algorithm to identify the correct objects in images without human intervention (Cashman et al., 2018, p. 40). YOLO uses these image segmentation features to beat their competition.

YOLO even beats out other types of algorithms used by Ege et al. (2018) themselves to estimate calories in dishes of food. Takumi Ege and Keiji Yanai hold a much broader place in research than just assessing the YOLOv2 algorithm. Their research covers bases from augmented reality to even social networking. In another paper by Ege and Yanai, the algorithm of choice was a CNN known as VGG-16. This algorithm proved quite effective at the task of computing calories within a dish, performing close to YOLO. More importantly, VGG-16 beat out DeepLab, further cementing CNNs as the neural network of choice for multi-image detection (Ege et al., 2017, p. 373). Despite contributing to CNNs superiority over other neural networks,

other features contribute in addition to backpropagation. An excellent feature of YOLOv2 specifically comes in the form of the way it manages regions within an image.

Figure 5

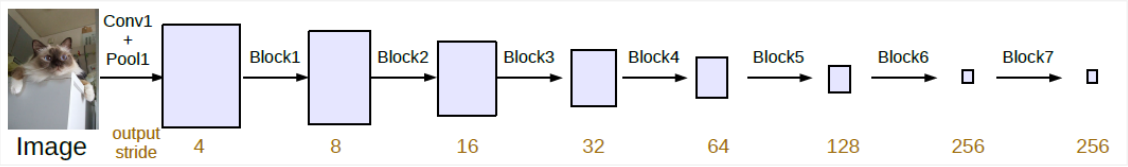
YOLOv2 Object Detection



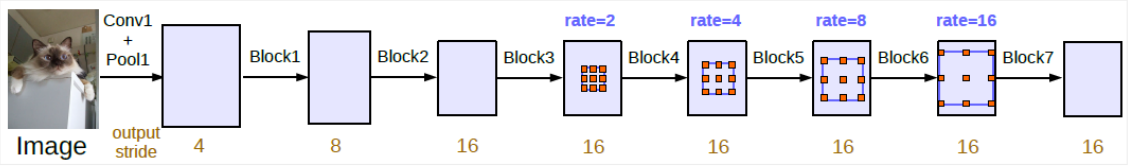
Contrary to networks like R-CNNs, YOLO looks at the complete image when determining objects, allowing quicker operating speeds unmatched by competition. Figure 5 shows a simplified version of the YOLO algorithm and how it detects objects (Redmon et al., 2016, p. 780). It simply splits the image up into bounding boxes, and then determines the chances of each box containing an object. This takes far fewer steps than the DeepLab algorithm as it does not have to run through the process of atrous convolution.

Figure 6

Comparing Atrous Convolution to Normal Convolution



(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

A large reason for DeepLab's slower operating speeds come in the form of one of its biggest strengths: atrous convolution. In Figure 6 (Tsang, 2019), we see the difference between the two types of convolutions in action. When going into deeper layers of a CNN, clarity diminishes due to a decrease image size. Atrous convolution counteracts this by adding artificial convolution parameters that maintain the image size throughout the network (Tsang, 2019). Despite adding clarity to the image, the change in performance does not appear drastic, and arguably comes in the form of a negative when accounting for speed.

Due to the simplicity of YOLOv2, it operates at a higher speed than DeepLab. According to Chen et al. (2015, p. 2), DeepLab's neural network operates at a speed of 8 fps (frames per second). YOLOv2 leads DeepLab by magnitudes in speed with a base network speed of 45 fps, hitting 150 fps on a quicker version of the network (Redmon et al., 2016, p. 779). For consistency both networks were tested on Nvidia TITAN GPUs. A quick network proves necessary in the landscape of competitive mobile applications where these algorithms reside and may result in a user's decision to keep using the application. Along with speed, these CNNs must ensure accuracy so the user receives correct results.

YOLOv2's biggest weakness comes in the form of accuracy. You only look once, YOLO's own name, makes the algorithm quick at detecting objects. It must only look at an

image once and it returns a result. This lets YOLOv2 outperform most other networks, including DeepLab. This quick image resolution results in lower accuracy, even slightly lower than that of RNNs. In the PASCAL VOC dataset, a collection of images containing everyday objects, both algorithms scored high among their respective competitors according to a leaderboard by paperswithcode.com, with YOLO falling slightly behind. This decrease in accuracy proves negligible in real world application due to YOLO's dominance in multi-object detection.

One of the largest differences in the capabilities of DeepLab and YOLO comes in one's inability to identify multiple objects within an image. Because of YOLO's simple nature, it only needs to look at an object once before conducting the process of object detection. A loss of accuracy occurs with such a quick algorithm. This allows YOLO to detect many different objects within an image. In the case of detecting calories within a plate of food, YOLO's multi-object detection proves essential for marketable software.

Social Impact

The availability of calorie tracking software greatly helps people on a weight loss journey. Many Americans lack the knowledge of caloric content in food which causes them to overeat in massive quantities. Easily palatable meals from fast food restaurants allow citizens to devour thousands of calories within minutes, far above the needs of a healthy person, while disregarding essential nutrients. Some people acknowledge the fact that they consume this food, aware that they have an addiction, but the target demographic of these algorithms comes in the form of people who do not even realize the quantity of calories they consume, and people who actively want to lose weight.

For health-conscious individuals, an effortless way to track calories emerges as invaluable. Many people simply have too many responsibilities for them to spend time counting

their calories, and others do not even know how to measure them. An app such as MyFitnessPal delivers an excellent starting point by allowing users to input foods that they eat and even scan barcodes in order to count the number of calories they consume in a day. The issues that arise with tracking calories, as mentioned previously, comes in the form of people not possessing the nutritional education to input values themselves.

With a quick and accurate algorithm, the problem of people not knowing calories vanishes. An ideal algorithm takes the guess work out of people's minds and speeds up counting calories. In the challenging task of counting all the calories in a large American dish of food, YOLOv2 accomplishes the job effortlessly. Thanks to YOLOv2's previously mentioned multi-image detection, an individual trying to stick a strict caloric diet must simply take a picture of their food and obtain a reading of its caloric content in mere seconds. The goal with the calorie counting problem is benefiting the most people, and YOLOv2 does a far better job than DeepLab v3+.

In addition to the usability of algorithms, the state of open source shows up as an additional ethical concern. Google invented DeepLab V3+, whereas YOLOv2 is open source. YOLO offers much more community support if a problem arises, but the end user must also rely on the creators to take time out of their day to fix the issue. Google, in contrast, has a massive team of developers paid and ready to fix potential problems that arise.

Large corporations like Google often have a catalogue of ethical concerns, such as tracking and selling user data. Some users and app developers strongly oppose practices that prey on the public. YOLO therefore beats DeepLab as a more ethical option for tracking calories and simply proves to be the better option in general.

Conclusion

DeepLab v3+ and YOLOv2 are two capable algorithms in the field of deep learning, especially when used for estimating calories in a dish, but YOLOv2 is the more efficient algorithm. YOLO therefore requires fewer resources and can perform its job quicker than DeepLab with competing levels of accuracy. This advantage can help combat widespread obesity by showing people that may be unaware how much they are really eating. For this reason, YOLOv2 is clearly the better algorithm as it has the most potential to defeat the problem of obesity by estimating calories quickly and accurately. The findings of this research can be profoundly useful in the topic of deep learning as this paper offers a broad view on the comparison of two convolutional neural networks that both show promise in tackling real-world problems like calorie estimation. These CNNs can be used for a multitude of other fields from finding breast cancer in its initial stages to analyzing weather patterns to predict natural disasters before they happen. Perhaps one day these networks will have the same functionality as a human brain, and maybe even more power.

References

An intuitive guide to Convolutional Neural Networks. (2018, April 24). FreeCodeCamp.Org. https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

Arora, D., Garg, M., & Gupta, M. (2020). Diving deep in Deep Convolutional Neural Network. *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, (pp. 749–751). Greater Noida, India: IEEE. Retrieved from https://doi.org/10.1109/ICACCCN51052.2020.9362907

Biswal, A. *Convolutional Neural Network Tutorial*. (2022). Simplilearn.Com. Retrieved April 19, 2022, from https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network

Cashman, D., Patterson, G., Mosca, A., Watts, N., Robinson, S., & Chang, R. (2018). RNNbow: Visualizing learning via backpropagation gradients in RNNs. *IEEE Computer Graphics and Applications*, *38*(6), 39–50. https://doi.org/10.1109/MCG.2018.2878902

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). *Semantic image segmentation with deep convolutional nets and fully connected CRFs*. *ArXiv:1412.7062 [Cs]*. http://arxiv.org/abs/1412.7062

Ege, T., & Yanai, K. (2017). Image-based food calorie estimation using knowledge on food. Categories, Ingredients and Cooking Directions. *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 367–375. https://doi.org/10.1145/3126686.3126742

Ege, T., & Yanai, K. (2018). Multi-task learning of dish detection and calorie estimation. *Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and*

Multimedia Assisted Dietary Management.(pp. 53-58). Stockholm, Sweden: ACM. Retrieved from https://doi.org/10.1145/3230519.3230594

Farsal, W., Anter, S., & Ramdani, M. (2018). Deep Learning: An overview. *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, (pp. 1–6). Rabat, Morocco: ACM. Retrieved from https://doi.org/10.1145/3289402.3289538

Gandhi, R. (2018, July 9). *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms*. Medium. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Hales, C. M. (2020). *Prevalence of Obesity and Severe Obesity Among Adults: United States, 2017–2018*. *360*, 8.

Johnson, D. (2020, February 10). *Back Propagation Neural Network: What is Backpropagation Algorithm in Machine Learning?* https://www.guru99.com/backpropogation-neural-network.html

Khandelwal, R. (2018, October 18). *Convolutional Neural Network(CNN) Simplified*. Medium. https://medium.datadriveninvestor.com/convolutional-neural-network-cnn-simplified-ecafd4ee52c5

Lewis, T. G., & Denning, P. J. (2018). Learning machine learning. *Communications of the ACM*, *61*(12), 24–27. https://doi.org/10.1145/3286868

Okamoto, K., Yanai, K., Adachi, K. (2021). Region-based food calorie estimation for multiple dish meals. *Proceedings of the 13th International Workshop on Multimedia for Cooking and Eating Activities* (pp. 17-24). New York, New York: ACM. Retrieved from https://doi.org/10.1145/3463947.3469236

Papers with Code—PASCAL VOC 2007 Benchmark (Object Detection). (n.d.). Retrieved April 19, 2022, from https://paperswithcode.com/sota/object-detection-on-pascal-voc-2007

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

TED. (2015, March 23). *How we teach computers to understand* pictures | Fei Fei Li [Video]. YouTube. https://www.youtube.com/watch?v=40riCqvRoMs

Tsang, S.-H. (2019a, March 20). *Review: DeepLabv3 — Atrous Convolution (Semantic Segmentation)*. Medium. https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74

Tsang, S.-H. (2019b, March 20). *Review: DeepLabv3 — Atrous Convolution (Semantic Segmentation)*. Medium. https://towardsdatascience.com/review-deeplabv3-atrous-convolution-semantic-segmentation-6d818bfd1d74

What causes obesity & overweight? (n.d.). Https://Www.Nichd.Nih.Gov/. Retrieved April 13, 2022, from https://www.nichd.nih.gov/health/topics/obesity/conditioninfo/cause