DeepNutrition: Food recognition from image for facilitating the dietary assessment process

Ismet Okatar
Department of Computer Science
Hacettepe University
Ankara, Turkey

Muhammed Aydogan
Department of Computer Science
Hacettepe University
Ankara, Turkey

Ali Kayadibi
Department of Computer Science
Hacettepe University
Ankara, Turkey

Dietary assessment plays a significant role in fighting obesity. In this paper, we used computer vision techniques to facilitate the dietary assessment process by detecting foods from images. We used the Food-101 dataset which consists of 101 classes and developed a model that uses a fine-tuned model of InceptionV3 and get 85% top-1 accuracy.

In this process, we tried different Models, Activation Functions, Loss Functions, Dropouts, Learning Rates, Batch Sizes. We observed the model's behavior in each of those conditions and tried to get the best case possible. Our results show that the InceptionV3 which is a CNN model can be a good choice for identifying Foods by using Food101 Dataset.

1 Introduction

In 2015 obesity caused 4 million deaths. Therefore it is an important condition in our modern humanity. People can easily overeat without knowing how much calories cost food. There are developed many ways to fight obesity. One of them is making a dietary assessment by using apps. There are applications to prevent that by measuring daily calorie intake by waiting from users to enter each of what they eat one by one. However, it can be a hard process to type every single food for lazy peoples. In traditional apps, it can take up to 72 seconds to type a meal.

The long process of input reduces the sustainability of dietary assessment. But by using computer vision techniques the time to record a meal reduces significantly to 9 seconds. Our aim is to facilitate that process and expect from users just to capture the photo before eating that meal. By that way capturing the photo can easily become a habit and people can raise their awareness in their diet with less volition.

Dietary assessment makes it possible to develop the consciousness of users and psychologically motivate them to limit their daily calorie intake. But it is still used traditional apps widely. Dietary assessment is not important for just counting daily calorie intake. Daily nutrition intake can also have some connections with other diseases. And it can be a great data source and lead to other health applications which uses the nutrition intake data and suggest different health solutions just from the mobile phones in our everyday pockets.

There is a lot of prior research in that area. There are ready datasets like Food101, UEC100, UEC256 which provide a big simplicity. We chose the Food101 dataset in this paper. We observed the behavior of the InceptionV3 model in various cases by changing the hyperparameters like Loss Functions, Optimizers, Activation Functions, etc. And we have observed some cases outperformed compared to other functions in our Food101 dataset

2 Related Work

As we mentioned before there is a lot of prior work done in that area before. The vision lab in ETH Zurich [1] has collected and labeled 101'000 images belonging to 101 classes each with maximum 512*512 pixels. It is also split to 750 train and 250 test images. The dataset is not perfect like having wrong labeled images to handle more edge cases in real usage.

In the paper [12] we can see that CNN out performed other traditional ways to detect foods from images. So we decided to use CNN instead of other machine learning techniques. But there is a lot of CNN Models which have strengths and weaknesses for certain conditions.

Yunsheng Ma clearly stated in [10], an algorithm to make food recognition systems is essential. In that way, developers can implement this somewhere in their app to make the calculation of dietary caloric intake. Since the ultimate

aim is finding a better way to recognize foods from a picture, we searched for different algorithms.

Lately, lots of studies about food detection from an image had published. One of them is [12]. Chang Liu's team has proposed a Convolutional Neural Network (CNN)-based algorithms with a few major optimizations, such as an optimized model and an optimized convolution technique. They combined LeNet-5 [13], AlexNet [14], and GoogleNet[15]. In [13], they tried to make multiple level neurons like the human brain, very closely. They use a 32x32 grayscale image. After several layers of convolution and sub-sampling generating a feature map and feeding it into the two fully connected layers, then after this layers' activation, giving values to 10 classed output layers. That was the basic CNN. In [14] five Convolutional and 2 fully connected layers are used with large scale labeled image data.

In [12], they say an input size of 224x224 taking RGB channels, with "1x1", "3x3" and "5x5" convolutions, yield the best result with 22 layers with GoogleNet. As clearly stated in [11], using small filters with several layers is way better than using big filters with one layer in order to decrease the parameter numbers.

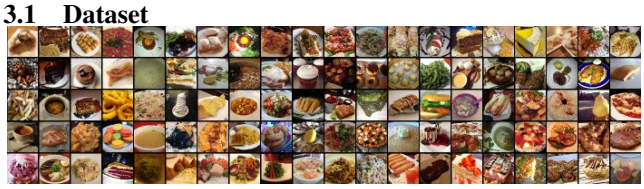
In [16] tests about VGGNet clearly shows VGGNet supports up to 19 layers. Built as a deep CNN, VGG also outperforms baselines on many tasks and datasets outside of ImageNet. VGG always shows a good average performance in every case. This is the reason which makes this algorithm very common and useful still.

In [26] we can see that Inception model gave 70.60% top-1 accuracy which is better than AlexNet which gave 56.40% top-1 accuracy and Discriminative Components with Random Forests which gave 50.76% top-1 accuracy. We can see that using ImageNet weights can be useful because there is a lot of similarity between ImageNet and Food101 Dataset.

Instead of starting from the scratch we tried to use those prior knowledge and develop our ideas in that base.

3 The Approach

3.1 Dataset

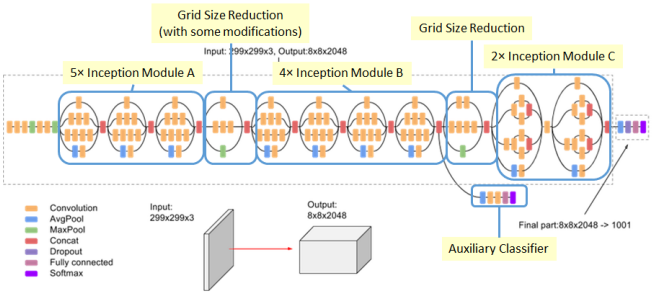


Food-101 Dataset

We have used Food-101 Data Set in our project.Food 101 has 101 food categories that has 1000 images for each food category.Images in each class are separated as 250 test images, 750 training images.There are some noise in the training images that is left on purpose.Noises can be intense colors or wrong labels.All images are 512x512 and the entire data set is 5GB.

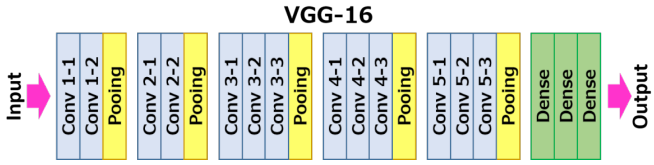
3.2 Model

CNN is a type of neural network and it is a image classification technique that is within Deep Learning and it is most often applied to image processing problems in Computer Vision applications. Convolutional neural network uses multiple layers. CNN has convolutional layers, pooling layers and sub-sampling layers followed by fully-connected layers. In 2015 ImageNet Large Scale Visual Recognition competition one of the CNN models which is Inceptionv3 become the first runner for image classification with 48 layers.It is the third version of Deep Learning Convolutional Architecture of Google. Inceptionv3 is trained on ImageNet dataset which has 1000 classes and 1 million training images.The network has an image input size 299x299.



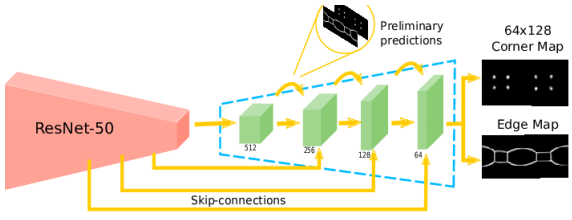
Inceptionv3 Architecture, taken from [20]

VGG-16 is another convolutional neural network model proposed by K.Simonyan and A.Zisserman. The model achieved 92.7% in ImageNet challenge. It is improved version of AlexNet by replacing large kernel sized filters with multiple small kernel sized filters one after another. But the model is slow to train and weights are large.



VGG-16 Architecture, taken from [17]

ResNet-50 is another convolutional neural network model.50 is number of layers in that model. What is specific about Resnet is it skips the connection. As we know from our lecture, when neural networks backpropagate, the gradients gets smaller. Small gradients make learning harder.The skip connection allows model to pass the input without entering weight layers. This allows to skip the layers that are less important in training and also offsets the gradient.



ResNet-50 Architecture taken from [19]

In our project we have decided to pick a model and than try to improve that model to make that model perform better.We have used 3 different CNN models which are Incep-

tionv3, VGG-16, Resnet-50 and we have decided to pick the one that gave the most validation accuracy. After our experiments with these 3 models, we have decided to pick Inceptionv3 which gave 79.08% accuracy. Our proposed model is improved version of Inceptionv3 and it gives 86.42% accuracy for our problem.Our model gives 7 percent more than what we had before. We have improved the accuracy of Inceptionv3 by using:

1. Transfer Learning
2. Data augmentation
3. Fine Tuning
4. Changing Hyperparameters.

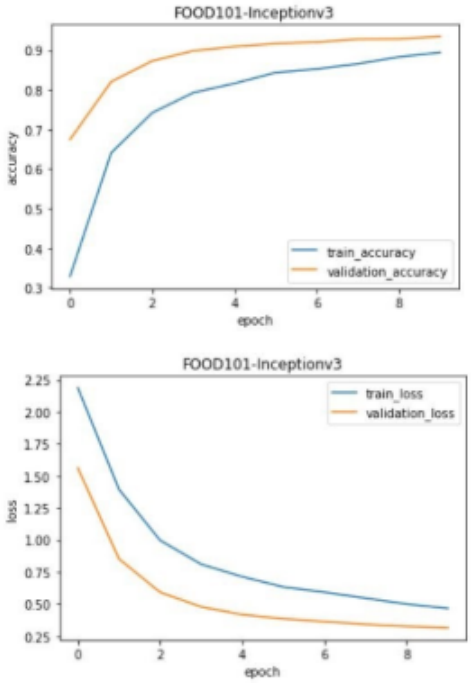
We will talk about them in the methodology part.

3.3 Methodology

In our project we have used Google Colab, Tesla K80 GPU to train our model. We have used TensorFlow, Keras. We have used Transfer Learning in our model. Transfer learning is using the pre-trained model and continue training on that model in another task. This helped us get faster results which saved a lot of time compared to models trained from scratch. We have used a pre-trained model of Inceptionv3 trained on ImageNet.This is the reason why accuracy starts high on our model. Then we fine-tuned our model. We have changed the last softmax layer of the Inception model and added a few new layers and a softmax layer that is relevant to our problem. Because the pre-trained model comes with 1000 categories on the softmax layer but in our problem, we have 101 categories. In our project, our softmax layer is 101 categories instead of 1000 categories. After that, we have done little data augmentation for our images. Food 101 is an already balanced data set but to improve the performance of our model we have used Data augmentation. We have used Sheering, zooming, and horizontal flip for our images. This can be improved more to get more accuracy in the future. And finally, we have done hyperparameter tuning in our project. We have experienced the effects of different Dropouts, Activation Functions, Learning rates and Momentums, Batch Sizes, Loss functions. Our train on the full model was taking more than one hour for each epoch so we have tried these on mini data sets and at the end, we have used every optimal parameter, full data set on our final model.

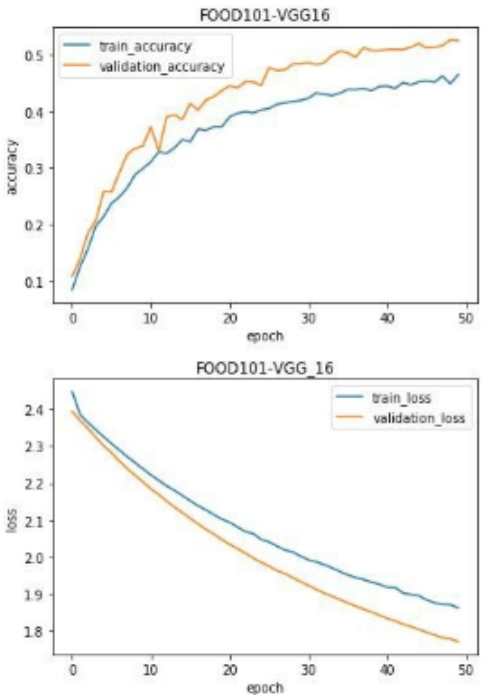
4 Experimental Results

Our first experiment was trying different models. For this we have tried 3 different models(Inceptionv3, VGG-16, Resnet-50). We got best results using Inceptionv3 model and we have decided to improve that model in our project. (These experiments are run on mini datas because it was taking more than one hour per epoch using full data.) This is the result we get from Inceptionv3.



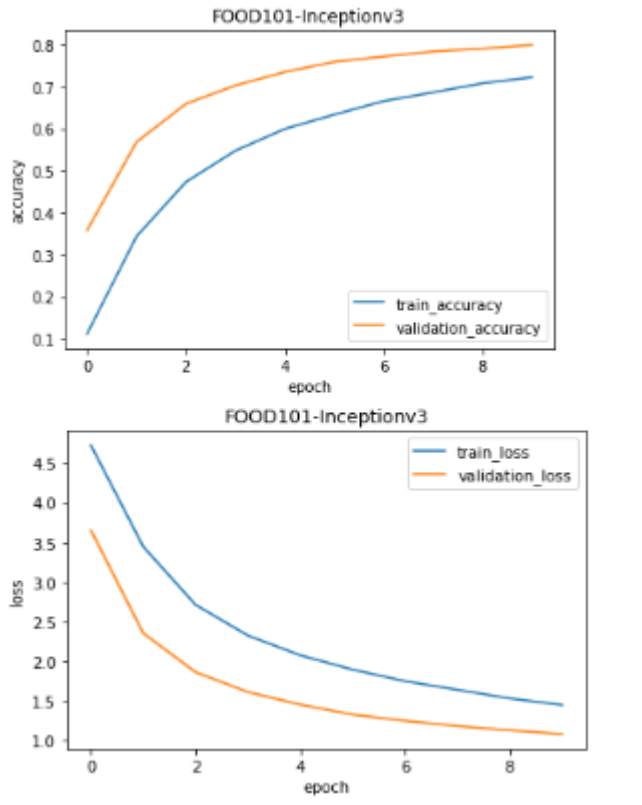
Inceptionv3 with 11 class

This is the result we get from VGG-16.



VGG-16 with 11 class

As we see from our experiments run on different models Inceptionv3 gave better results compared to others. So we wanted to use Inceptionv3 model to improve current performance. This is the output we get from Inceptionv3 model using our full data.

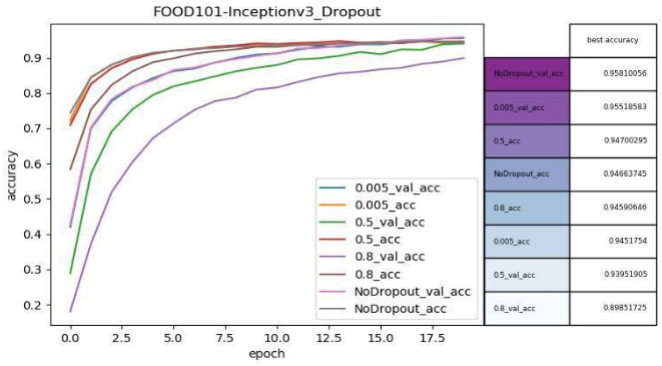


Inceptionv3 with 101 class

We have trained our model for more than ten hour and this was the output we got using 101 classes.

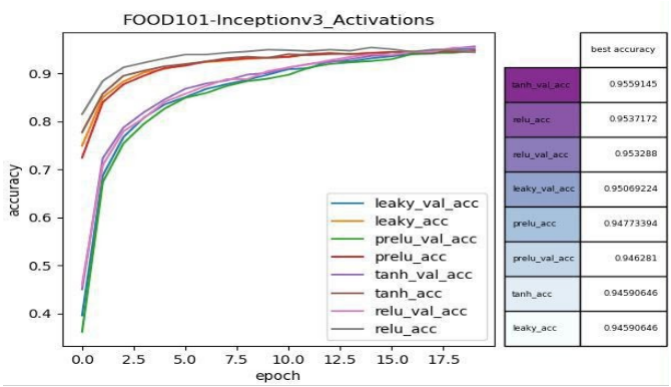
After selecting the model, we wanted to see the effects of different hyperparameters such as Dropouts, Activation Functions, Learning rates and Momentums, Batch Sizes, Loss functions. To experience this we have trained our model multiple times with different parameters. We have changed one parameter and kept the other same to see the effect of that particular parameter. These experiments are run on mini data because it was taking more than one hour using our full data. So we have used the results of mini data to improve our model trained with full data.

Firstly, we have tried different Dropout values and no Dropout. Dropout is generally used to avoid overfitting. It ignores randomly selected neurons during training so that other neurons handle the representation to make predictions. By doing that our network becomes less sensitive to specific weights of neurons and it generalizes better. In our model, we got the best accuracy using no dropout. Our dataset was a balanced data set so this might be the reason why using no dropout gave better accuracy.



Inceptionv3 with different Dropout values

Secondly, we have tried different activation functions. For this we have used leakly val, prelu, tanh, relu functions. Activation functions are used to introduce non-linearity to our models,which allows our models to learn nonlinear boundaries. In our case, best activation function was tanh function.



Inceptionv3 with different Activation Functions

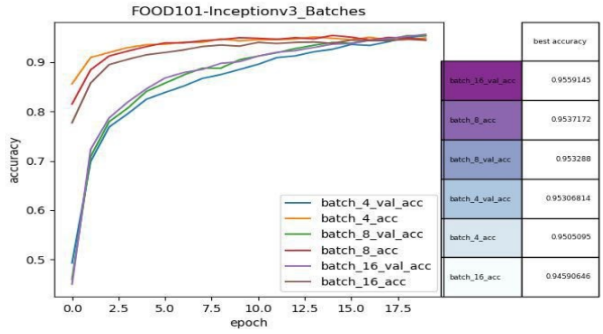
Thirdly, we have tried different learning rates with and without momentum. Learning rate is amount that weights are updated during training. If we select low learning rate, our training process would be long and if we select large learning rate, we may miss global minimal, or model learns weights too fast and does not update weights later. Momentum is size of the steps taken towards the minimum and it shows the direction towards the minimum from the previous steps. In our experiments we have tried 0.0001,0.001,0.0005 learning rates with and without momentum and we got the best result using 0.0005 learning rate with 0.9 momentum. And using momentum with same learning rates always give better results, this means that momentum is useful in models.

Learning Rate	Momentum	Train Accuracy	Validation Accuracy
0.0001	0.9	97.74%	94.30%
0.0001	0	92.63%	82.09%
0.001	0.9	99.43%	95.21%
0.001	0	94.73%	94.23%
0.0005	0.9	99.06%	95.29%
0.0005	0	89.40%	94.08%

Inceptionv3 with different Learning Rates and Momentum

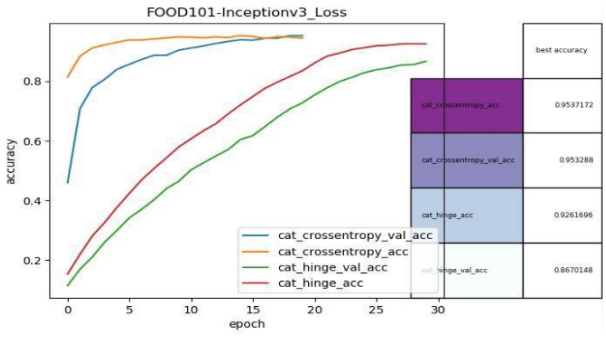
Fourthly, we have tried different batch sizes for training our model. Batch size is number of sub-samples given to network. We tested 8 (succesfull), 16 (succesfull), 32 (failed),

24(failed), 20, 18, 17 failed. Then we decided that 16 is the maximum we can work with, due to lack of GPU memory.



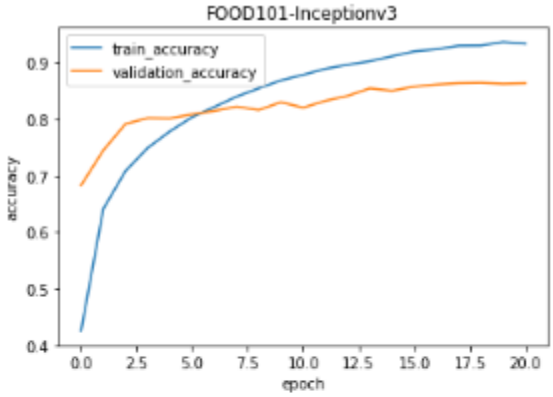
Inceptionv3 with different Batch Sizes

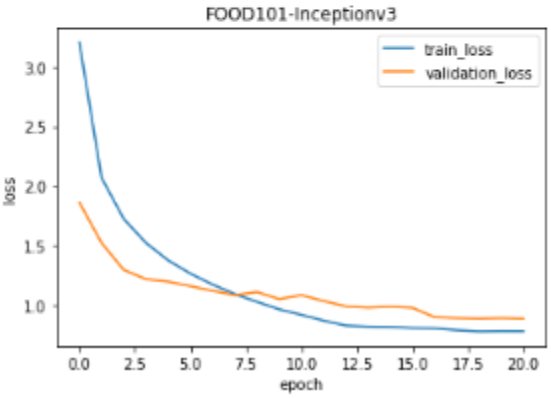
Fifthly, we have tried different loss functions. Loss functions are methods of evaluating how well our algorithm models our dataset. We got the best results using categorical crossentropy.



Inceptionv3 with different Loss Functions

Finally, we have combined our work to a single model to see if we achieved our aim. We have trained our model with full data and with transfer learning, fine tuning,data augmentation and hyperparameter tuning. After training our model more than twenty hour, we have achieved 86.42% accuracy , we had 79.08% before so we can say that we have improved performance of our model.





Inceptionv3 with Optimal Hyperparameters

And we have tested our model with different food images we got from Nefis Yemek tarifleri.

Our models performance with example foods,taken from[25]

Our model was successful in predicting these images. But our model also has weaknesses, our model confuses similar-looking foods such as filet mignon and steak(one is cut before cooking other is cut after cooking), chocolate mousse and chocolate cake, dumpling and gyoza(gyoza is Japanese dumpling), apple pie and ice cream(because of the shape of the glass). But one of our major weaknesses is our model classifies non-food images as well. For example for my personal image model classified me as a churro so in the future our model can be extended with food - non-food classifier.

5 Conclusions

We made experiments to gain more accuracy for recognizing foods in order to facilitate the dietary assessment process for fighting obesity. We observed various models and parameters behaviors at Food101 Dataset which contains 101 classes and 101'000 images. We have seen that using the InceptionV3 model outperformed compared to VGGNet and ResNet. And using tanh activation compared to ReLu, PReLu, LeakyReLu gave a better result in our current case. Setting Dropout to 0.0 instead of 0.8 gave much better accuracy. We can see that not just the model but choosing the right hyperparameters is also an important factor for getting high accuracy.

We can say that using CNN models when dealing with images gave a huge advantage. And using pre-trained weights like ImageNet weights can speed up the training process and gave a huge simplicity for us because we were lack of GPU.

In the future, we will work with more powerful GPUs which will let us make our experiments faster and wider. So we can test our model and our dataset with many various HyperParameters, Optimizers, Activation Functions, Loss Functions, etc. We also want to experiment on our dataset's behavior by using CNN Fusion and Boosting methods. Due to lack of GPU, we were unable to make our experiments in full 101 class in Food101 Dataset. Instead of that, we used a minimized version that contains 11 classes and 11'000 images 224*224 each. It has accelerated our model's training time and gave us the ability to make many more experiments with various hyperparameters.

Due to every layer is connected to every layer before DenseNet is very memory hungry architecture. But it has outperformed compared to every other methods and it is now identified the state of the art in the food recognition area. Due to the lack of GPU, we were unable to make experiments and observe the behavior of the DenseNet architecture. After InceptionV3 Architecture it would be a great choice to make further experiments on DenseNet architecture.

We also want to try different augmentation methods in order to deal with some separating problems. Due to some meal dishes have the same color and contains a huge area in the total of the image it can be a hard problem to separate them. So there can be used different augmentation methods to deal with that situation.

After all, we reached 85% top-1 accuracy in our Food101 dataset we reached that by experimenting. This is a great achievement for us. By making further improvements in that area in future the food recognition can be used in our daily lives more and by facilitating the dietary assessment process the consciousness for nutrition intake can be increased. And not just for fighting with obesity but maybe it can lead to other health applications that use the nutrition intake data. And it can make our lives better and healthier and extend our lifetime.

References

[1] Food101 https://www.vision.ee.ethz.ch/datasets$_extra/food-101$

[2] Chang Liu , Yu Cao, Yan Luo, DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment, The University of Massachusetts Lowell, One University Ave, 2014

[3] https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats

[4] https://vitalab.github.io/article/2017/03/16/vggnet.html

[5] https://www.tensorflow.org/api$_docs/python/tf$

[6] https://keras.io/

[7] https://www.pyimagesearch.com/start-here/

[8] https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848

[9] https://www.cs.toronto.edu/ frossard/post/vgg16

[10] Yuji Matsuda, Hajime Hoashi and Keiji Yanai, "Recognition of Multiple Food Images by Detecting Candidate Regions"

[11] Review: VGGNet — 1st Runner-Up (Image Classification), Winner (Localization) in ILSVRC 2014 https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11

[12] Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane, Yunsheng Ma, "DeepFood: Deep Learning-based Food Image Recognition for Computer-aided Dietary Assessment"

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE vol. 86, pp. 2278-2324, 1998.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in NIPS, 2012, p. 4.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[16] Jerry Wei, "VGG Neural Networks: The Next Step After AlexNet" https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-alexnet-3f91fa9ffe2c

[17]https://neurohive.io/en/popular-networks/vgg16/

[18]www.researchgate.net/profile/Clara$_Fernandez$13$/publication/$3273

[19]https://www.quora.com/What-is-the-deep-neural-network-known-as-"ResNet-50"

[20]https://www.mathworks.com/help/deeplearning/ref/inceptionv3.htm

[21]https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

[22]https://medium.com/@bakiiii/microsoft-sunar-deep-residual-network-d2970003ad8b

[23]https://www.techopedia.com/definition/32731/convolutional-neural-network-cnn

[24]https://www.youtube.com/watch?v=K$_BHmztRTpA$

[25]https://www.nefisyemektarifleri.com

[26][https://pdfs.semanticscholar.org/6dbb/4f5a00f81971b7bc45f670f37