Intelligent Fruit Maturity Assessment Platform Using Convolutional Neural Network: IFMAP

Jihene Rezgui, Thomas Trépanier, David Génois, Hervé Claden and Amasten Ameziani
Laboratoire **R**echerche **I**nformatique **Ma**isonneuve (LRIMa)

Montreal, Canada
jrezgui@cmaisonneuve.qc.ca, thomtrepanier@gmail.com, david.genois13@gmail.com, hervecladen@gmail.com, aliamasten@gmail.com

Abstract—As the fight against climate change and the environmental crisis continues, the need for solutions to reduce our carbon footprint increases dramatically. In this paper, we propose a platform to help prevent food waste among Canadian households using an innovative artificial intelligence (AI) system that implements Convolutional Neural Networks (CNN) to analyse pictures of fruits taken with cell phones to determine their current maturity state. Following these results, we have created a native mobile application for the platform which is detailed in the article. We also explain the overall workflow of the platform as well as the different neural networks used and tested for this task. Afterwards, we discuss the use of a wide range of sensors to give more insights about the fruit to our neural networks. We then show a use case of this application on a banana, which was analysed accurately by the neural networks. Finally, we display different results we obtained for the CNNs such as the top performance of different architectures, which can classify the data up to the accuracy of 97.11%. We also demonstrate that a CNN (LeNet) training time is three times faster on a GPU compared to a CPU.

Keywords—Food Waste; Neural Network; Fruit; Food Maturity, Sensors, Mobile application;

I. Introduction

Food waste is a problem we have tried to solve for a long time. The need for a solution grows bigger every day as it becomes more important than ever to solve this problem. Every year, Canadian homes throw away 2.2 million tons [1] of food. Fruits and vegetables makeup around 45% of this amount. In Canada, it accumulates to a loss of around 7.65 billion dollars per year. As mentioned, almost half of the food wasted by Canadian homes are fruits and vegetables. This can be attributed to the fact that, most of the time; we buy our products fresh and mature and never eat them before they perish. The difficulty to evaluate if a fruit is good for consumption or the inability to make use of it before it perishes, whether it is by forgetting about it or not knowing how to use it are among the causes for such food waste.

Even though a fruit or vegetable is past its expiry date, it does not mean it is dangerous. Canadians tend to throw away food that does not look appealing or appetizing. Consequently, when they forget to eat a fruit, it will be thrown away instead of being used in a specific recipe that could make use of such an item.

We think the artificial intelligence paradigm can be used to create a solution to this problem. After doing a literature review of existing solutions and comparing them together as explained in the next section, we came up with the idea to create our novel platform, called "IFMAP." With our IFMAP algorithms, we provide a way to utilize neural networks to decrease food waste. We have achieved improvements in algorithms discussed in prior research in ways that are explained in the following sections.

Our contributions in this paper can be summarized as follows: (1) We introduce our Intelligent Fruit Maturity Assessment Platform, called "IFMAP" [2] which can identify fruits and their level of maturity using Convolutional Neural Network (CNN); (2) We propose ways to utilize this algorithm to help consumers be more informed using a mobile application. We also explore the possibility to re-use the same algorithm in many areas of the food supply chain such as processing or manufacturing; (3) We introduce the IFMAP architecture composed of four vital components: The neural networks, the server, the mobile application and the alternative sensor; (4) We demonstrate the effectiveness of the improvements made with IFMAP heuristics by analyzing various approaches we experimented with and (5) We developed two variants of IFMAP's workflow which can accommodate different levels of complexity expressed by working with a single or double CNN configuration.

Section II provides a brief overview of related work and tools to compare them with our platform. Section III contains a thorough description of our platform and its different components. Section IV present use cases our platform is currently able to handle. Section V shows IFMAP's simulation results, and Section VI concludes our paper.

II. Related Work

We have done a literature review to find and compare existing solutions as seen in Table 1 below. We have looked at both recognition and analysis tools. According to our findings, we believe a different solution is needed to help consumers waste less food, as each existing product did not treat every aspect of the problem we are targeting even though it was effective at what it is meant to do.

Our review of the literature showed that many systems to reduce food waste already exist during the production and transportation part of its journey. Such systems are the Centaur

978-1-7281-5628-6/20/$31.00 ©2020 IEEE

[3] initiative: a whole platform dedicated to monitoring the quality of cereals during transportation and storage through AI. Other solutions that implement AI to reduce food waste are sales forecasting neural networks [4] which are trained on the massive amount of data of customers' buying habits to learn to predict future sales in stores and restaurants. Companies can then order the appropriate amount of food to minimize waste. However, as mentioned, those solutions are not aimed at the customers and that is what we want to remedy.

Table 1: Related work summary

Solution	**Product Summary**	**Comments**
FoodHero—Fight Food Waste & Save Money [5]	A mobile application that shows deals on items grocery stores are about to throw away.	—Does not provide image analysis tools, or information on already purchased products -Useful to reduce food waste at the distribution phase.
AgShift [6]	Analyser that utilizes machine vision and deep learning to provide insight into a product's defects.	—Does not apply to the average consumer. —Works on a few selected items for now. —Good accuracy
Centaur [3]	Platform handling food quality during the post-production phase with the use of olfactory sensors. Mostly used for grains.	—Does not apply to the average consumer. —Useful to reduce food waste before food reaches the consumer.
Sales Forecast in E-commerce using Convolutional Neural Network [4]	Algorithms to predict the quantity of food a grocery store will sell daily.	—Does not apply to the average consumer. —Does not provide an analysis of the image of already purchased products. —Good planning tool for grocery stores

These findings guided us in our process to try and find a way to approach food waste at home. Despite the existence of tools to identify fruits like *Google Lens* [18] or tools to analyze the state of a fruit like *AgShift* we found a lack of affordable tools that managed to be efficient in both areas. The detection of the fruit is a part of the process, but the analysis of it is also important. This led us to the concept of IFMAP as detailed in the next section.

III. IFMAP ARCHITECTURE

In its current state, the platform is made of four vital components: (A) The neural networks, (B) the server, (C) the mobile application and (D) the alternative sensor as shown in Fig.1.

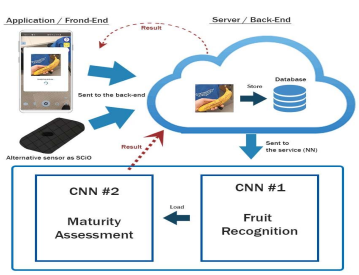

Fig. 1 IFMAP Workflow

This workflow has different advantages across all the main components. The exact workflow of each component is described further in this section. The two main advantages of having two different CNNs for the fruit recognition and fruit maturity assessment are: 1-Flexibility and 2-Expansion Capability. First of, with two different networks, we can use different and more specialized data that is adapted either to fruit recognition or maturity assessment to get better results. We can also pinpoint which part of the analysis works as expected and which one is deficient to improve the system. This flexibility also allows us for instance to recognize a higher number of fruits than the number that we can evaluate the maturity of since the data for the later task is quite scarce. Second, the expansion capability means when we will want to be more precise in the maturity evaluation (e.g. a weekly precision instead of a Fresh or Rotten evaluation), we will have a specialized network for this task. In fact, the main point of having a different network is for us to input more parameters such as the room's temperature, texture or even input from a different sensor (see [14]) which are essential to give an accurate evaluation of a fruit's maturity level. These parameters, however, should not affect the fruit recognition and they might confuse the network when it is trying to identify the fruit if a single network is used. Therefore, having a separate network enables us to draw on these parameters to improve our maturity assessment precision while not affecting the identification part of the process.

Furthermore, having full control over the application and server side of the platform provides us with many advantages. This includes ease of expansion, analytics capabilities and improvements to the user experience. In other words, according to our findings and needs, we will be able to collect data on specific user activities in order to better identify the user's needs to improve the platform and to help advance our research project. This collected data can also help to identify potential user behaviors which could yield useful ideas or confirm theories. We see a lot of potential in the data the platform could provide us with.

A. *Neural Network*

The Neural Network (NN), which will be referred to as *service*, is the primary tool in the IFMAP platform. It was built with the DeepLearning4J library [7] and the machine we used for the training has a **non-overclocked Intel Core i5-9660K** CPU, **16 GB of RAM** and the GPU is a **NVIDIA GeForce GTX 1060 6GB** from Gigabyte. The service's purpose is to analyse the picture taken and evaluate the maturity level of the fruit presented. This analysis works in three phases which can be described as follows: (1) recognizing the fruit, (2) assessing the fruit's maturity and (3) compiling and sending the results back to the server. It is to note that we use two different NN (referred to as the double configuration) for the fruit recognition and evaluation. As of now, this service is able to assess apples, bananas and oranges and categorize them as either "Fresh" or "Rotten" with a level of precision of 97.11% (Table 4 and 5). The particular difficulty of this task is that the AI has to recognize a rotten apple and a fresh apple as the same fruit and then be able to extract enough data from the picture to accurately decide whether it is rotten or not.

First of all, when the service receives a picture, it crops it to the appropriate size for the first NN to use (in our case, 100x100 pixels). The first NN is what is called a Convolutional Neural Network (CNN), which is a type of Neural Network that is notorious for being good in image recognition and image classification. We modelled our CNN#1 on AlexNet's architecture [9] (Fig. 2.A) which we found to be the most successful architecture with our dataset among other models like the LeNet [8], a SimpleCNN [10] architecture and the SqueezeNet [11]. The AlexNet model also has the advantage of maintaining a rather small size and fast computing time, which, in the case of a mobile application, are valuable qualities. We also tested the VGG16 [12] and ResNet [13] architecture, but those ended up being too large and too slow for our needs. This model's architecture is illustrated below:

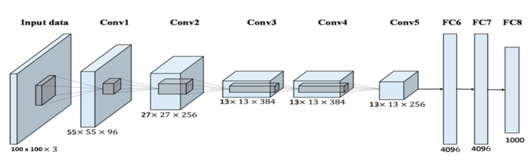

Fig. 2. A AlexNet's model architecture

Second, the service uses the answer of the first NN to locate and load the appropriate CNN (CNN#2) to evaluate the fruit's maturity. However, if the accuracy of the first prediction is below a certain threshold, we cut the process short because the prediction has a high chance of not being right and return a blank answer to the server to save time. We tried different values for the threshold and found that a value of 90% was the most accurate for the platform. If the accuracy is above that threshold, we consider that the prediction is viable, and we can continue with the fruit maturity assessment. The CNN#2 loaded is modelled after LeNet's architecture which performs similarly to the AlexNet architecture, but is significantly faster to execute and smaller to store. These performances are compared in Table 5. This model's architecture is illustrated in Fig. 2.B.

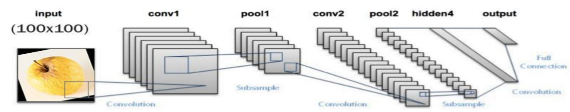

Fig. 2.B LeNet's model architecture

Before the picture is passed through this second CNN, we brighten and increase the contrast in it with the *RescaleOp* class of the Java.Awt package with a rescale factor of 1.5 and an offset of 30. We found that it increased the real-world accuracy of IFMAP, since a lot of real pictures were dimly lit compared to those in the datasets used in training. After adjusting the picture, we pass it through the evaluation CNN which outputs either a "Fresh" or "Rotten" status for the fruit.

Finally, before we send the results to the server, we verify once again that the second CNN has a high enough accuracy for the result to be viable. We use a different threshold of 70% for this comparison. If the accuracy is above that value, we send both the fruit's name and its maturity status to the server with the respective accuracy. However, if it is too low, it only sends back the fruit's name and its accuracy.

<u>Data Cleanup</u>

Our dataset originally contained 10,901 samples, but many of them were not appropriate for the type of analysis we wanted to perform. Such samples were either obstructed by inappropriate objects, duplicates or contained multiple fruits in one picture (Fig. 3. A and 3.B) which wasn't the kind of data we wanted to train on.

Fig. 3. An uncommon banana disposition for the CNN#2 to evaluate

Fig. 3.B Stack of oranges whose maturity can't be properly assessed

Therefore, we ended up with a smaller dataset of 9503 (Fig. 4) samples. We combined our fresh and rotten images of each fruit for the NN to learn to recognize them and split them again when it was time to train the NN to differentiate the two.

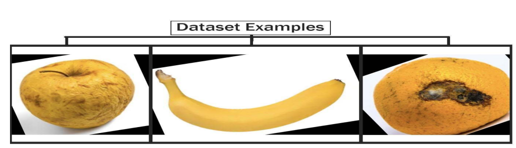

Fig. 4. Here is an example of the dataset that was used (From left to right: Apple-Rotten, Banana-Fresh, Orange-Rotten)

B. *Application*

Other than the Neural Network component, which is the main tool our platform uses to identify fruits and their maturity level; the platform's application is composed of a modern hybrid mobile app built on the React Native framework [15]. This component will be referred to as the frontend component. The frontend and neural network components are linked together by the server-side application as mentioned previously. It will be explained in more detail in the next bullet point.

The current application is aimed at consumers, but can be adapted to fit the business's needs, in either case, the objective and means remain the same: to reduce food waste using a platform that utilizes neural networks. The frontend side of the platform is the first component of the application users interact with. Therefore, it plays a critical role in the user adoption rate of the platform. This is the basis of our core design goal to provide an intuitive user-friendly environment where the end user of the platform can scan fruits and then receive information based on the results.

The identification of the fruit and its maturity could be used to provide the user with various information on his fruit to make better-informed decisions. Examples of information which can be given to the consumers to help them make informed decisions in the short-term and long-term are how to potentially identify if a fruit is good by analyzing the texture or knowing how to conserve it more efficiently. It is important to keep in mind that identifying a fruit's maturity precisely only with an image is, to the best of our knowledge, not possible. Many aspects of a fruit's current state are invisible to the naked eye and require sensors which consumers typically do not have [14] as explained in **Section III.D**. This explains why we would not provide a business with the same information as a consumer. If a user has a sensor (both business or consumer) which can provide more data about a fruit, then a more accurate result could be given. We believe it would be unsafe to provide a user with a so-called accurate maturity assessment by only analysing a picture.

In the future, we would like to be able to provide more information on where a product is from and to allow the use of a barcode scanner to use the platform for things other than fruits. Another important feature we will work on is a kitchen manager section to keep the application's ecosystem relevant to the user. This would allow reminders to be set to prevent users from forgetting about expiring items.

As we move forward, many of those functionalities we intend to implement will require a free account to access them. It is important to us to keep the current state of the application available to everyone without locking it behind a mandatory account for ease of access. This is what we call guest mode. Anyone who uses the platform will be able to try it by scanning a fruit before signing up.

*C. **Server***

The backend server stands as the core of our project's flow. It serves the purpose of allowing the neural networks and the application to communicate together and gives us more flexibility as to how we will handle the data.

Its middleman position allows us to filter the requests which then open multiple possibilities to us that we will detail further in this part.

The architecture of the project and its data flow are as follows: the user of the application has the choice to either send a picture as a guest or as a logged in user. Once the picture is sent to the server, a service worker running the neural network component as a service is called and the picture's path on the server is sent as an argument. The results are sent to the backend who will register them in the database with the relevant timestamps and send them back to the application. The end user gets to know what he captured, its freshness and the neural networks' degree of certainty of the answer once the results are received.

The stored results in the database allows us to have a better grasp of the neural networks' evolution and, through the history, double-check the results to correct its behavior.

It also gives interesting statistics for further purposes on the volumes of specific vegetables consumption in given areas and depending on when and how often a user will use the application will allow us to define a person's shopping habits, for example. The statistics are, at the moment of the document's writing, still in development, but it is definitely something we will seek pursuing. In the future, it will give us the opportunity to give better advice to the logged in consumers as to how to manage their grocery shopping to avoid waste.

As of now, the communication between the application and the neural network component is fully functional. The history registration in the database works well but at the state our platform is at, we've decided not to log the user's location data yet even though the boilerplate for that feature is set. We wanted to focus on the fruit maturity assessment and how to optimize the process as detailed in the use case in **Section IV**.

*D. **Alternative sensor (SCiO)***

Our platform is ready to easily integrate an alternative sensor. Using alternative sensors to increase the maturity evaluation accuracy and precision are part of IFMAP's workflow. Such sensors could take the form of an ultraviolet sensor, a hardness sensor or a molecular sensor like SCiO's [14] consumer-friendly sensor which can analyse a fruit's molecular construction to give insights into the fruit's current state. This information can then be sent to the server in conjunction with the picture taken and fed to the service (CNN#2) to improve the evaluation and give a more precise result.

Connecting such a sensor to our platform and mobile application would take the form of an IoT project on which we discuss further in the conclusion (**Section VI**).

Furthermore, as this is a more expensive variant than our original solution, it might not be available in every household. However, companies and warehouses can use this technology to complete and confirm their evaluation which is a very important step in the production process. This sensor could also be bought by grocery stores and markets to ensure and prove the quality of their product.

IV. Detailed Use Case

In its current experimental form, to access and test the platform, it is required to go to the remote repositories and download the 3 core components of the platform [2]: Backend, Mobile Application and IFMAP_NN_Service_2.

Fig. 5. A Camera screen

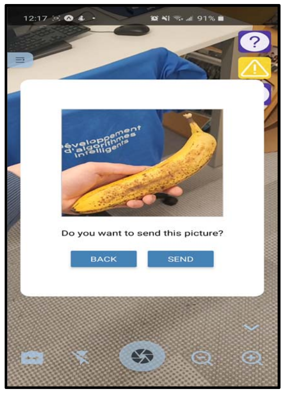

Fig. 5.B Confirmation modal

Fig. 5.C Sending image to the backend and fetching results

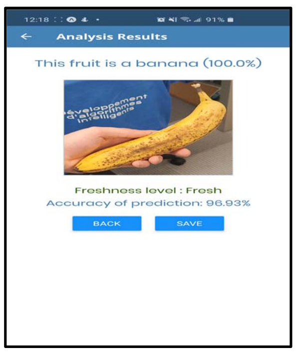

Fig. 5.D Result screen with data acquired from the backend

Each of the component's repositories contains a detailed "readme" file, which explains how to get the component to run as expected.

When the platform is running and once the user enters the app, they need to be connected to the internet—or local network in this case. They can take a picture of a fruit or upload a picture to receive information from the platform directly from the main screen. Here follows a typical use case.

In this example, let's say our test user just bought a banana and wants to know how to store it in the most optimal conditions. From the main page (Fig. 5.A), the user just needs to line up their fruit in the white frame on the screen, and then press the camera button at the bottom. When the picture is sent to the backend, a spinner appears while the image is sent to the server and then analysed by the NNs (Fig. 5.C) as explained in **Section III.A**. After the treatment is complete, the response sent back by the backend is forwarded to the detail screen, where the cropped picture will be shown once again, but this time with the information received from the server telling us what our fruit is, and how fresh it is. (Fig. 5.D)

V. Simulation Results

A. Neural Network Parameters

The following simulation results were obtained using, if not specified otherwise, the parameters of Table 3.

Table 3. Neural Networks Parameters

Parameter	**Value**
Image size	100 x 100 pixels
Number of channels (CNN)	3
Seed (Randomizes shuffle)	123
Batch size	50
Number of epochs	50

Table 4. Top Fruit Recognition Accuracy (CNN#1)

Name	**Accuracy**
AlexNet	90.96 %
LeNet	88,34 %
SimpleCNN	68,38 %
SqueezeNet	31,05 %

Table 5. Top Maturity Assessment Accuracy (CNN#2)

Fruit	**LeNet**	**AlexNet***
Apple	97.19%	98.29%
Banana	100%	99.82%
Orange	97.61%	98.73

*Image size for AlexNet was 224 x 224 pixels

B. Single Configuration vs. Double Configuration

Table 6: Accuracy comparison of a single/double configuration

Number of CNNs	**Overall accuracy**
Single CNN: AlexNet	97.11 %
Double CNN: AlexNet + LeNet (IFMAP's technique)	88.49 %

A big part of IFMAP's workflow is the division of work between the fruit's recognition and its maturity assessment, each part being the job of a specific network. However, we also tried to train a single model to do both jobs at the same time. We trained this network with the same data as before, but labelled the data differently (e.g. fresh-apple, rotten-banana) such that the model would both recognize the fruit and evaluate its maturity in one pass or one network (referred to as a single configuration). In fact, the power of CNNs and neural networks in general is their ability to understand complex problems and concepts at a high level without our need to dissect it for them. We decided to use two CNNs (double configuration) in the first place with IFMAP because we thought that it would be too difficult for a single network to appropriately recognize a fruit and accurately determine its maturity.

However, since our problem is much simpler at this time (i.e. Fresh vs. Rotten), the task can also be done with a single CNN. For that case, AlexNet's architecture demonstrated a higher performance than the other aforementioned architectures. A comparison of the accuracy of both configurations is shown in Table 6.

As we can see, a single CNN using AlexNet's architecture outperforms a double CNN configuration by a margin of 8.62%. However, as explained at the beginning of **Section III**, we didn't use a single model configuration to have better flexibility and expandability with IFMAP once the problem's complexity increases or if we want to dive deeper into the analysis of a specific fruit.

C. *CPU vs. GPU training time*

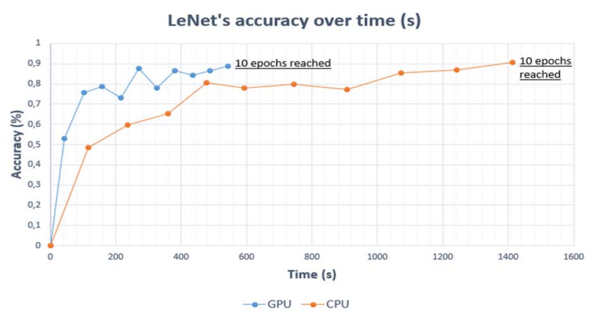

Fig. 8. CPU vs. GPU training time using LeNet's architecture to detect and evaluate the fruit maturity.

The DL4J library allowed us to choose either to train our networks on the CPU or the GPU of the computer. This is implemented using NVIDIA's CUDA [16] and cuDNN [17] libraries. We decided to compare the performance of both methods in the time it took to train a network. This CNN recognizes and evaluates the fruit at the same time (single configuration) with the LeNet's architecture. We trained the models with the same parameters for 10 epochs and the results are illustrated in Fig. 8.

As expected, the GPU trains the network much faster than the CPU can: it took, on average, 54 seconds to complete an epoch on the GPU while the CPU took 120s. Furthermore, we found that the CPU and the GPU handle computation very differently. For instance, we were not able to train the AlexNet models with a CPU as it seemed that their accuracy stagnated at 26.25%.

VI. Conclusion

In conclusion, there are many opportunities for growth and data to be acquired in all the components of the platform. For the application itself, we want to publish it to the various online stores to begin acquiring user data and expand its functionalities to improve the user experience. This includes a virtual refrigerator, and the ability to set reminders to help users waste less food. For the neural network component, we want to increase the accuracy of our maturity assessment model to a weekly precision. As of now, we have not collected enough data for this task to be possible, but we hope that our users feed us enough data to train such a neural network. Finally, as mentioned earlier, we are aware that only using a picture to determine a fruit's maturity is not enough. Therefore, we want to increase the number of inputs the CNN#2 takes into account to give a more accurate and realistic estimate of a fruit's time before it expires. Such inputs would come from external sensors [14] connected to the platform which would send additional information about the fruit's state to the service to help it achieve the desired precision. Once these goals are achieved, we want to expand the number of fruits our platform can assess and potentially be able to evaluate some vegetables too.

References

[1] "Food Waste in Canada". Lovefoodhatewaste.com, 2018. https://lovefoodhatewaste.ca/about/food-waste/ [last visited on 13/01/20]

[2] IFMAP platform by Thomas, David, Herve, Amasten and J. Rezgui on GitLab, https://gitlab.com/ifmap [last visited on 04/01/20; open source code.]

[3] D.L. Brabec et al. "Evaluation of Wireless Phosphine Sensors for Monitoring Fumigation Gas in Wheat Stored in Farm Bins". ISSN 2019, 2019 https://www.researchgate.net/publication/332733629_Evaluation_of_Wireless_Phosphine_Sensors_for_Monitoring_Fumigation_Gas_in_Wheat_Stored_in_Farm_Bins [last visited 22/01/20]

[4] K. Zhao and C. Wang "Sales Forecast in E-commerce using Convolutional Neural Network", 2017 [last visited 22/01/20]

[5] "The only Québec-based app to fight food waste—FoodHero is now available at nearly 200 IGA stores across Québec". www.newswire.ca, 2019. https://www.newswire.ca/news-releases/the-only-quebec-based-app-to-fight-food-waste-foodhero-is-now-available-at-nearly-200-iga-stores-across-quebec-823245576.html [last visited 22/01/20]

[6] Sainul Abudheen K "AgShift has developed an autonomous food inspection system to reduce global wastage". e27.co, 2018. https://e27.co/agshift-autonomous-food-inspection-system-tackle-global-wastage-20180914/ [last visited 22/01/20]

[7] DeepLearning4J official website. https://deeplearning4j.org/ [last visited 13/01/20]

[8] Y. Lecun et al. "Gradient-Based Learning Applied to Document Recognition". Proceedings of the IEEE 1998 [last visited 13/01/20]

[9] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". NIPSC [last visited 13/01/20]

[10] SimpleCNN Algorithm by Octavio Arriaga on Github, https://github.com/oarriaga/face_classification/ [last visited 13/01/20]

[11] Forrest N. Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size". ICLR 2017 [last visited 13/01/20]

[12] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". ICLR 2015 [last visited 13/01/20]

[13] K. He et al. "Deep Residual Learning for Image Recognition". IEEE CVPR 2016 [last visited 13/01/20]

[14] M. Li et al. "Evaluating the performance of a consumer scale SCiO™ molecular sensor to predict quality of horticultural products" ScienceDirect, 2018. https://www.sciencedirect.com/science/article/abs/pii/S0925521418302990 [last visited 22/01/20]

[15] React Native framework by Facebook inc. https://facebook.github.io/react-native/ [open source framework] [last visited 13/01/20]

[16] Khoirudin and J. Shun-Liang. "GPU APPLICATION IN CUDA MEMORY", ACIJ, 2015. http://airccse.org/journal/acij/papers/6215acij01.pdf [last visited 22/01/20]

[17] S. Chetlur et al. "cuDNN: Efficient Primitives for Deep Learning". 2014. https://arxiv.org/abs/1410.0759 [last visited 22/01/20]

[18] Achom Nishalakshmi Devi, Gaurav "Reviews on Augmented Reality: Google Lens". International Journal of Computer Trends and Technology (IJCTT) V58(2):94-97, April 2018. ISSN :2231-2803. www.ijcttjournal.org. Published by Seventh Sense Research Group. [last visited 22/01/20]