UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA
CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO

Publicación de Artículo en Congreso Internacional con Arbitraje o en Revista Indizada

Visual Recognition with Machine Learning using Cloud services

QUE PARA OBTENER EL TÍTULO DE:

MAESTRÍA EN INGENIERÍA DE LA COMPUTACIÓN.

P R E S E N T A :

ING. MARTHA ADRIANA SORIANO MÉNDEZ

ASESOR:

DR. JUAN CARLOS CUEVAS TELLO

SAN LUIS POTOSÍ, S.L.P, SEPTIEMBRE 2019

FACULTAD DE INGENIERÍA

A mi madre, abuela y hermanos por su amor, paciencia y apoyo incondicional en el desarrollo de este trabajo.
A Juan, por sus palabras y confianza para terminar exitosamente esta etapa.

Acknowledgment

To my advisor Dr. Juan for the correct orientation, support, patience and collaboration.

To the postgraduate of Computer Engineering for the unconditional support for the presentation of the article in ISUM 2019.

Resumen

El reconocimiento de imágenes es utilizado para realizar una gran cantidad de tareas, en las cuales el objetivo es encontrar contenido o patrones en las imágenes. Los algoritmos de aprendizaje profundo realizan los mejores resultados para este tipo de problemas. La red neuronal convolucional, CNN por sus siglas en inglés, es lo último en tareas de clasificación. Sin embargo, a veces requieren de mucho tiempo para obtener un buen rendimiento, especialmente con una gran cantidad de datos. En este caso, la paralelización es la mejor solución para disminuir el tiempo utilizado durante la etapa de entrenamiento. Este trabajo evalúa diferentes tecnologías de aprendizaje profundo con el mismo conjunto de datos para comparar el tiempo de entrenamiento y la precisión. Las tecnologías evaluadas son: i) MATLAB: Máquinas de vectores de soporte (SVM) Naive-Bayes y algoritmos de aprendizaje profundo (AlexNet y GoogLeNet, arquitecturas CNN). ii) IBM Cloud, Visual Recognition. iii) Keras, en modo secuencial y paralelo. Los mejores resultados son fueron obtenidos por IBM Cloud, GoogLeNet y AlexNet con una precisión superior al 90%, donde el más rápido fue AlexNet. La versión paralela con Keras mejora significativamente el tiempo de entrenamiento.

Abstract

Image recognition is used to perform a large number of task with the objective to find content or patterns in images. Deep learning algorithms performs the best results for this kind of problems. Convolutional Neural Network (CNN) is the state-of-the-art on classification tasks. But sometimes it requires so much time for getting a good performance especially with a huge amount of data. In this case parallelization is the best solution to decrease the time used during the training stage. This work evaluates different deep learning technologies on the same dataset in order to compare the training time and accuracy. The technologies evaluated are: i) MATLAB: Support Vector Machines (SVM) Naive-Bayes and deep learning algorithms (AlexNet and GoogLeNet, CNN architectures). ii) IBM Cloud, Visual Recognition. iii) Keras, in sequential and parallel modes. The best results are from IBM Cloud, GoogLeNet and AlexNet with an accuracy above 90%, where the fastest is AlexNet. The parallel version with Keras improves significantly the training time.

Contenidos

Contents

List of Figures

Chapter 1

Acceptance Letter

SOMEXSU A.C.
Sociedad Mexicana de Supercómputo

ISUM
www.isum.mx

Guadalajara. Jal, 28 de febrero 2019

Dra. Martha Adriana Soriano Méndez
UASLP
P r e s e n t e .

En nombre del Comité Académico Nacional del 10th International Supercomputing Conference in México (ISUM 2019), el cual se realizará del 25 al 29 de marzo en Monterrey, N.L., me es grato informarle que el trabajo con el tema **"Visual Recognition With Machine Learning Using Cloud Services"** que sometió para presentar en el evento, fue aceptado para una ponencia.

Podrá ver en la página del congreso www.isum.mx la hora y el día de su ponencia. Favor de poner atención al programa que se publique en la página.

También, le recordamos que si está interesado en publicar su trabajo necesitamos que envíe su trabajo en extenso a más tardar el 25 de marzo del 2019.

Sírvase esta carta como constancia de su próxima participación en el ISUM 2019.

Si tiene alguna pregunta favor de enviar un correo a papers2019@isum.mx. De nuevo felicidades y nos vemos en el ISUM 2019 en Monterrey, N.L.

A T E N T A M E N T E

Dr. Moisés Torres Martínez
ISUM National Committee, Chair

Chapter 2

Attendance Certificate

Figure 2.1: Certificate ISUM 2019

Chapter 3

Article

Visual Recognition with Machine Learning using Cloud Services

Martha Adriana Soriano-Mndez*, Gabriela Rangel-Ramirez*+, and Juan Carlos Cuevas-Tello*

*Faculty of Engineering, Computer Engineering UASLP; +Instituto Tecnolgico Superior de San Luis Potos, Capital
adrianasorianom@gmail.com,cuevas@uaslp.mx
gabriela.rangel@tecsuperiorslp.edu.mx

Abstract. Image recognition is used to perform a large number of task with the objective to find content or patterns in images. Deep learning algorithms performs the best results for this kind of problems. Convolutional Neural Network (CNN) is the state-of-the-art on classification tasks. But sometimes it requires so much time for getting a good performance especially with a huge amount of data. In this case parallelization is the best solution to decrease the time used during the training stage. This work evaluates different deep learning technologies on the same dataset in order to compare the training time and accuracy. The technologies evaluated are: i) MATLAB: Support Vector Machines (SVM) Naive-Bayes and deep learning algorithms (AlexNet and GoogLeNet, CNN architectures). ii) IBM Cloud, Visual Recognition. iii) Keras, in sequential and parallel modes. The best results are from IBM Cloud, GoogLeNet and AlexNet with an accuracy above 90%, where the fastest is AlexNet. The parallel version with Keras improves significantly the training time.

Keywords: cnn, parallelization, keras, machine learning

1 Introduction

The industry has evolved since the first industrial revolution in England (1760-1870), which was lead by mechanical production – mainly steam engines. The second industrial revolution (1870-1970) was driven by electric power and oil. The third industrial revolution, started in the mid-twentieth century, is characterized by electronics and computers. Nowadays, we are living the fourth industrial revolution, known as Industry 4.0.

This new industry concept was introduced by Germany, and there are recent technological advances: i) Big Data and Data Analysis, ii) Cloud Computing, iii) Cybersecurity, iv) Robotics, v) Internet of Things, vi) Simulation of computer processes, vii) Augmented Reality, viii) Autonomous robots and ix) 3D Print Process integration.

The main objective of Industry 4.0 is to build an "intelligent factory", where intelligent machines exchange information with each other and adapt themselves.

Consequently, the experts expect to obtain a rapid increase in productivity and significant energy and material savings. That is why automation is an important concept in Industry 4.0.

The automotive industry is the one with the most growth in automation [17]. This industry continues to be a sector that faces challenges such as regionalization, saturation of some markets, globalization, technological advances, new competitors and its continuous restructuring. The main uses of automation occur during production processes, especially in areas such as welding, painting, chassis engraving and car body measurements.

Among the benefits of automation are the improvement of quality processes, standardization in manufacturing, reduction of material waste, cost reduction, increased production and security what leads companies to invest large amounts annually in computer science.

Computer science has contributed in an essential way to the automation of processes, whose main objective is to improve the efficiency of the company in relation to the expectations of the clients. In an automation process, the computer is the fundamental tool, and should be enhanced with the appropriate platform, all in line with the vision and corporate strategy at the computer level. Essentially the disciplines of computer science as big data, cybernetics, artificial intelligence, expert systems, augmented reality and mechatronics among others have been used to achieve the automation and all the benefits mentioned previously.

One of the areas of great interest for companies is the Artificial Intelligence (AI), which can be defined as the integration of electronic circuits and computer programs in order to emulate the functioning of the human brain and perform motor activities without human intervention. A well-planned and implemented AI generates great profits for the company. An example is the production lines, where the parts of the car are assembled through a band with sensors, which detect when the chassis has reached a certain point, a timer keeps the band stopped for a certain time, while the robot assemble or paint the piece. This type of automation allows companies to achieve production levels that would not be possible if the work were done by an employee. Also, reduce the margin of error that could have the employee for any reason, call fatigue, distraction, lack of training, etc.

A recent trend is to automate the inspection of products to comply with total quality processes, which preserve and reinforce the policies and organizational culture that the production systems that have been implemented for years in the automotive industry. One of the fields with a lot of opportunity in this line of research is the AI Vision. This field through the use of appropriate techniques, allows obtaining, processing and analysis any kind of special information obtained through digital images. Typical objectives of artificial vision include:

- The detection, segmentation, location and recognition of certain objects in images (for example, human faces).
- The evaluation of the results (for example, segmentation, registration).

- Registering different images of the same scene or object, that is, matching the same object in different images.
- Following an object in a sequence of images.
- Mapping of a scene to generate a three-dimensional model of the scene; This model could be used by a robot to navigate the scene.
- Estimation of three-dimensional human postures.
- Search digital images for their content.

It is in this field where the most important advances of the AI are being carried out. In practical terms, machine learning is the science that is responsible for making computers perform actions without the need for explicit programming. The main idea here is that you can provide data to the machine learning algorithms and then use them to know how to make predictions or guide decisions [13, 16].

Using machine learning can improve some automotive fields like advanced driving assistance systems, autonomous driving and others fields related with sales and after sales processes, creating applications with a high potential. Machine learning is used for process all the information obtained from sensors, cameras, etc., and it has been proven as a capable tool for detecting patterns in data and deriving predictions. But within machine learning a new technique has emerged, deep learning [10].

Deep learning refers to a set of machine learning algorithms that utilize large neural networks with many hidden layers for tasks, such as image classification, speech recognition and language understanding as examples [9]. Deep learning can help to organize data and improve the data collection process from visual inspection in manufacturing process. In social media analytics, deep learning can assist and improve data collection and analysis. In case of robots and smart machines, enables self-learning robots to become more intelligent over their lifetime.

BMW IT Research Center has been working in three AI applications in the automotive industry, the areas in which they have worked are: visual inspection of vehicles and parts (Visual Inspection), the automated detection of bar code labels on boxes to optimize the goods received process (Deep Logistics) and the detection of the trailer ID on trucks in the trailer yard (Deep Yard Management) [9, 10].

Two approaches have been studied for the Visual Inspection problem:

- Clemson University with BMW IT Research Center considers the problem of helping a human user setting up an automated Visual Inspection System (VIS). The VIS is intended to assist the human inspector in checking whether a part has been installed or not, or if there are different types of parts, to check if the correct part has been installed. This research focus on image similarity metrics such as i) specific average difference weighted normalized cross correlation and ii) cumulative difference weighted normalized cross correlation. Also they use a Gaussian Blur filter and the Sobel operator to extract the edges to help similarity metrics. The VIS system is not totally automatic since the operator must select manually where the parts

are located. For the test cases, some metrics work fine and others do not. They also reported that VIS is sensible to rotation, translation and scaling [15].
- Another research work on visual inspection, they mention the need to classify large number of images collected from hundreds of cameras in the plant in a central location. So far exist a dataset in the BMW Spartanburg plant, South Caroline USA, this dataset contains images from 4 vehicle types and 25 camera perspectives. It currently consists of 82,011 images. They employ pre-trained neural networks: GoogLeNet, AlexNet, and Inception. These nets were trained with ImageNet dataset [8]. They reported that the accuracy varies depending on category between 44% and 97%, and 81% is the average accuracy [9].

The contribution of this paper is the measurement of the performance from different machine learning algorithms for visual recognition through three different technologies: Deep Learning from MATLAB®, IBM Cloud[1] – AI – Visual Recognition and The Python Deep Learning Library known as Keras[2], running sequentially and in parallel.

The first part of this paper explain the dataset construction used in all the experiments. The second part show the methodology for each technology. And the last part of this paper presents the obtained results from our experiments.

2 Dataset

To build our dataset, we employ the Cars dataset [7] as a basis, and we do a new selection of these images and categorize them into new classes. The original dataset contains 16,185 images of 196 classes of cars, split in 8,144 images for training and 8,041 for testing. The problem was only the training images were labeled, so we can only work with the training set. So if we divide 8,144 images among 196 classes, then we have only 41 images for training and testing. Therefore, this number of images per class is insufficient to carry out a good experiment. That was the reason to reorganize the images into color classes. We group the images in 5 classes: black, blue, gray, red and white as shown in Figure 1. Each class has 300 images: 250 for training and 50 for testing (83.3% training - 16.7% testing) as shown in Table 1. This new dataset we call it Color Cars.

3 Methodology

To evaluate the performance of different Machine Learning algorithms as Naive Bayes [12] [5], SVM [4], and Convolutional Neural Networks (CNN) we used three different tools: MATLAB, IBM Cloud – AI– Visual Recognition and Keras.

[1] https://www.ibm.com/cloud/
[2] https://keras.io/

Fig. 1. Color Cars dataset

Table 1. Dataset

class	# training	# testing
blue	250	50
white	250	50
gray	250	50
black	250	50
red	250	50

3.1 MATLAB

MATLAB was used to test SVM [4], Naive Bayes [12] [5], AlexNet [8] and GoogLeNet [3] as CNN, using as the technique Transfer Learning. Transfer learning is a machine learning technique where a model developed for a past task is reused as a starting point to train a new model. One of the benefits of this technique is the performance increases training only the last layer or the selected layers of the pre-trained model with the new model information.

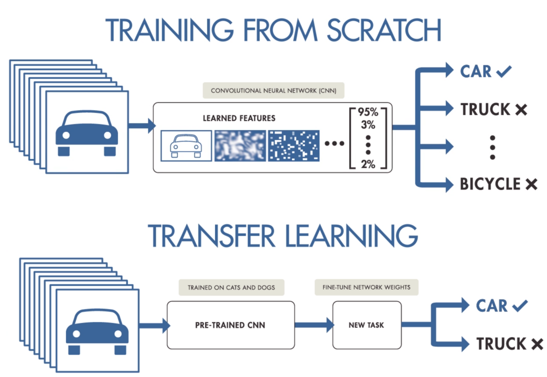


Fig. 2. Transfer Learning [11]

3.2 IBM Cloud – AI– Visual Recognition

In Visual Recognition IBM Watson we evaluate the performance of the service.Visual Recognition use deep learning algorithms to analyze images for scenes, objects, faces and other content. The experiment was carried out through IBM Academic Initiative [6] and agreement between IBM and schools, in this case with the Autonomous University of San Luis Potosi (UASLP). This initiative makes available to the university community a large amount of resources associated with IBM software, such as licenses for more than 1,200 IBM products. That implies knowing IBM services without a cost. Being a no-cost agreement has some disadvantages in terms of available resources. For example, in Visual Recognition service you can create only one custom model and each class has a limit of 250 images for training. This is the reason we limit our dataset to that amount of images. Once the model was trained with visual tool you can test it with the same tool or connect your application to the web service and process the information obtained to the desired format (Image 3). We developed a python application where user can select and image and test it with the trained model. The result will show the name of the class with higher result and its value.

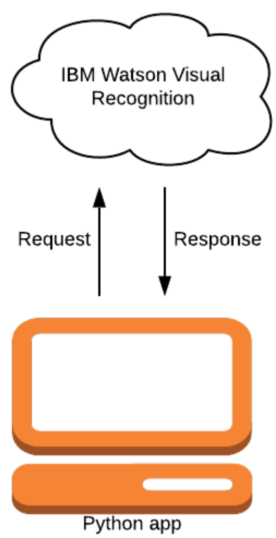


Fig. 3. Visual Recognition - IBM Watson

3.3 Keras

The main objective to execute some experiments using Keras framework was the possibility to implement parallelism and reduce the training time of a CNN from scratch, see Figure 4. The computer where parallelism was implemented has a Nvida GK110 GPU. This GPU has 2880 cores and the base clock run to 875 MHz. The complete specifications can be consulted in the following link [1]. We implemented a basic CNN and compare the training time using parallelism and running in sequential mode. Furthermore, to improve the performance of the model, we implemented data augmentation and compared the results.

```
:  #initialize the cnn
   classifier = Sequential()
   #step1 convolution
   #classifier.add(Convolution2D(32,(3,3), input_shape = (64,64,3), activation='relu'))
   classifier.add(Convolution2D(32,(3,3), input_shape = (64,64,3), activation='relu'))
   #pooling
   classifier.add(MaxPooling2D(pool_size = (2,2)))
   #step2 conv
   classifier.add(Convolution2D(128,(3,3), activation='relu'))
   #pooling
   classifier.add(MaxPooling2D(pool_size = (2,2)))
   #step3 conv
   classifier.add(Convolution2D(256,(3,3), activation='relu'))
   #pooling
   classifier.add(MaxPooling2D(pool_size = (2,2)))
   #step3 conv
   classifier.add(Convolution2D(512,(3,3), activation='relu'))
   #pooling
   classifier.add(MaxPooling2D(pool_size = (2,2)))
   #flattening
   classifier.add(Flatten())
   #full connection
   classifier.add(Dense(units = 128, activation = 'relu'))
   #activation function softmax
   classifier.add(Dense(units=5, activation = 'softmax'))
```

Fig. 4. CNN from scratch. Architecture used as baseline.

4 Results

Four experiments were performed using MATLAB, one for each algorithm: SVM, Naive Bayes, AlexNet and GoogLeNet. The dataset used is as described at §2. All experiments were done on the same computer to measure the computing time accurately. The computer specifications are CPU Xeon E5-2695 v2, 16 GB RAM non-ecc DDR3 with a mother board: Asus ROG rampage IV black.

The first experiment was performed with SVM. It took approximately 80 seconds to train. The matrix confusion is in Table 2. The accuracy measures the percentage of correctly classified images on test images.

Table 2. SVM - Accuracy = 66%

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	33	0	1	1	0
BLUE	3	24	18	0	2
GRAY	8	23	24	7	1
RED	6	3	7	40	3
WHITE	0	0	0	2	44

In the second experiment, Naive Bayes was used. The training time was 2,634.03 seconds and the accuracy is 62.8%, see the confusion matrix in Table 3.

Table 3. Naive Bayes - Accuracy = 62.8%

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	30	0	4	3	0
BLUE	3	27	12	1	4
GRAY	7	18	29	7	1
RED	10	5	5	35	9
WHITE	0	0	0	4	36

AlexNet was the third experiment, and it lasts 91 seconds on average during training, and the accuracy is 90.32% – see confusion matrix in Table 4.

Table 4. AlexNet - Accuracy = 90.32%

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	50	3	2	0	0
BLUE	0	45	0	0	0
GRAY	0	1	37	0	2
RED	0	0	1	50	0
WHITE	0	1	10	0	48

GoogLeNet was the fourth experiment, and it took 6 minutes and 30 seconds on average during training. The accuracy is 92% – see confusion matrix in Table 5

Table 5. GoogLeNet - Accuracy = 92%

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	50	2	3	0	0
BLUE	0	44	1	0	0
GRAY	0	3	41	0	4
RED	0	0	1	50	0
WHITE	0	1	4	0	46

For Visual Recognition from IBM Cloud, we uploaded five zip files to train our custom model (one zip file per class). Once the zip files had been uploaded correctly, the training phase started and took 57 minutes. The training is performed on the servers of IBM Cloud. The testing is made by sending an image to the custom model and the IBM Cloud returns a number between 0 and 1, a confidence value. Then we need to define a threshold to decide what class is accepted. Therefore, four experiments were carried out with different threshold value (0.9, 0.8, 06 and 0.5). The goal was to evaluate the accuracy for each threshold. A confusion matrix for each execution were generated, see Tables 6 to 9.

Table 6. Threshold 0.9 - Accuracy = 79.6%

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	48	1	0	0	0	1
BLUE	0	39	0	0	1	10
GRAY	1	0	14	0	2	33
RED	0	0	0	50	0	0
WHITE	0	0	0	0	48	2

Table 7. Threshold 0.8 - Accuracy = 86.8%

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	48	1	0	0	0	1
BLUE	0	43	1	0	1	5
GRAY	2	1	26	0	7	14
RED	0	0	0	50	0	0
WHITE	0	0	0	0	50	0

The 6^{th} column in Tables 6–9 contains the number of images that were not evaluated (positive value) or that were doubly evaluated (negative value). It depends of the assigned threshold. For example, when one image is evaluated with 0.5 in black and 0.7 in blue it will be doubly evaluated; in case the image get 0.6 as a highest result and the threshold for testing is set in 0.8, then it will

Table 8. Threshold 0.6 - Accuracy = 90.8%

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	49	1	0	0	0	0
BLUE	1	45	2	0	2	0
GRAY	4	1	33	0	13	-1
RED	0	0	0	50	0	0
WHITE	0	0	0	0	50	0

Table 9. Threshold 0.5 - Accuracy = 92.4%

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	49	1	0	0	0	0
BLUE	2	45	2	0	2	-1
GRAY	5	1	37	0	14	-7
RED	0	0	0	50	0	0
WHITE	0	0	0	0	50	0

not pass the evaluation. To understand the previous explanation and check our theory, we made an image by image test with 0.5 threshold. From Tables 6–9 we can see that the best results were obtained with the threshold set to 0.5. In fact, this value is the recommended by IBM Cloud documentation. The testing time on IBM Cloud is 237 seconds equivalent to 3.95 minutes. The best accuracy obtained was 92.4%.

Regarding the results from the Keras framework, we performed several experiments to evaluate the best performance against the number of epochs, see Figure 5. As the plot shows the best result was 44 epochs for this CNN, see §3.3. The solid line represents training accuracy and the dotted line the testing accuracy:

Another experiment carried out, it was to measure the training time by using Keras in sequential mode. It took 2879.9627 seconds, approximately 47.99 minutes. Compared with IBM Cloud – Visual Recognition service, it took 10 minutes less. The confusion matrix in sequential mode is in Table 10, and the accuracy is 86.8%.

Table 10. Keras sequential mode - Accuracy = %86.8

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	35	5	5	2	4
BLUE	2	45	2	0	2
GRAY	5	1	37	0	14
RED	0	0	0	50	0
WHITE	0	0	0	0	50

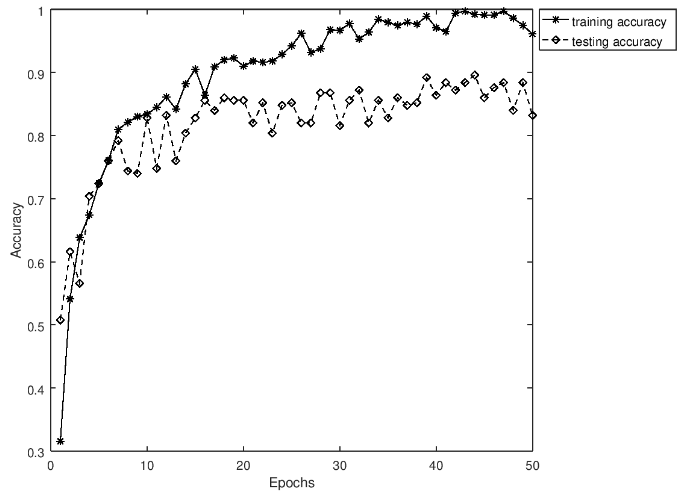


Fig. 5. Sequential performance of CNN with Keras. Number of epochs versus Accuracy.

The running time on Keras and IBM Cloud is time consuming, therefore a parallel approach is straightforward. Our next experiment that was carried out is to find the optimal number of workers to run in parallel by using Keras. We evaluate from 5 to 100 workers, where 15 workers was the optimal number, see Figure 6.

The confusion matrix and accuracy with 15 workers is shown in Table 11. As we can see, the accuracy does not differ so much with the accuracy obtained in sequential mode, see 10. The small difference is due the random initial weights of CNN.

Table 11. Keras parallel mode - Accuracy = %81.6

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	32	8	8	0	2
BLUE	0	47	1	0	2
GRAY	1	1	28	0	20
RED	0	0	0	48	2
WHITE	0	0	1	0	49

It is important to emphasize that the results shown in Table 11, data augmentation was implemented to generate the dataset during training. However when we remove data augmentation the results radically decreases, see Figure 8.

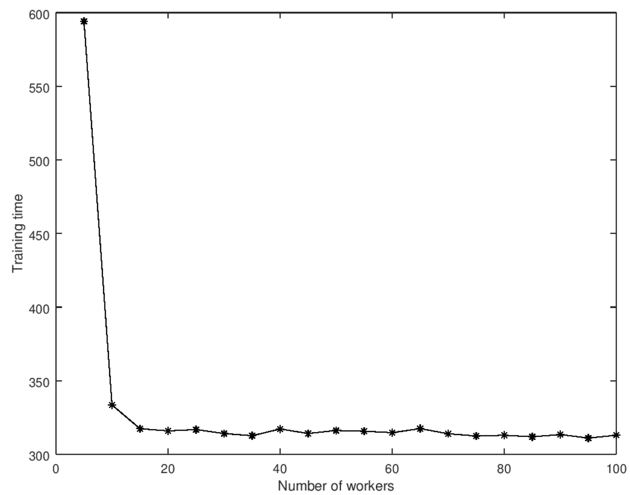


Fig. 6. Comparison of training time with different numbers of workers

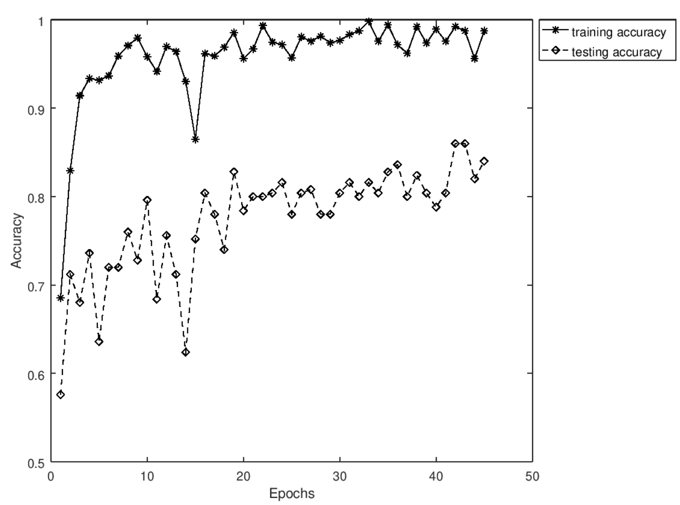


Fig. 7. Parallel execution with 15 workers and data augmentation

Finally, the Table 12 shows the summary results from all experiments, we compare the accuracy and the training time. The testing time on all technologies is similar.

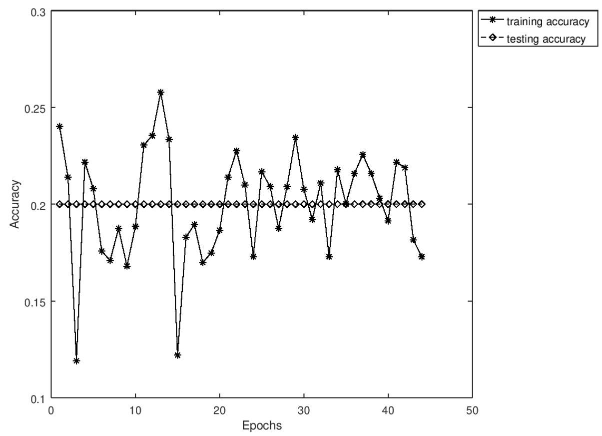


Fig. 8. Parallel execution with 15 workers without data augmentation.

Table 12. Comparison: Accuracy and training time

Methods	**Accuracy(%)**	**Training time**
SVM	66	80 s
Naive Bayes	62.8	43.9 min
AlexNet	90.32	91 s
GoogLeNet	92	6 min 30 s
Visual Recognition IBM Cloud	92.4	57 min
Keras (Sequential)	86.8	47.99 min
Keras (Parallel-Data augmentation)	81.6	5.28 min
Keras (Parallel-No data augmentation)	20	5.27 min

5 Conclusions and Future work

In this research we studied the image recognition problem. Companies are forced to invest in more efficient visual inspection systems, capable of detecting even the smallest failure during the production cycle, therefore computer vision techniques are so important in the detection process. For example, a computer vision system for the automatic detection of a correct positioning of doors and windows in a production process in a real environment is used [2].

There are several datasets for image recognition. In this paper, we employed our own dataset Color Cars, which was obtained from the dataset Cars, see §2.

We compared different machine learning algorithms with different technologies.The technologies evaluated are: i) MATLAB: Support Vector Machines (SVM) Naive-Bayes and deep learning algorithms (AlexNet and GoogLeNet, CNN architectures). ii) IBM Cloud, Visual Recognition. iii) Keras, in sequential and parallel modes.

The best results are from IBM Cloud, GoogLeNet and AlexNet with an accuracy above 90%, where the fastest is AlexNet. Nevertheless, GoogLeNet and AlexNet are pretrained with ImageNet and data augmentation, and training time is not taking in account. On the other hand, IBM Cloud is a blind architecture, since one does not know the hyper-parameters that were used. We conclude that they use some kind of CNN and data augmentation.

The parallel version with Keras improves significantly the training time. But more knowledge on machine learning to set up your model is required.

A problem was presented with the experiments developed with Keras framework. The accuracy decreases when data augmentation was eliminated due the small amount of images used for training. Even though same dataset was used on other frameworks (MATLAB and Visual Recognition IBM Watson) good accuracy was achieved on these frameworks, which leads us to think that possibly some image pre-processing is done internally.

Another big problem that CNNs has is that if they were never trained with rotations or translations on their training images, they have problems for correctly classify in the test stage. To solve this problem, researchers use the technique called data augmentation. This technique is used by the main CNN architectures, and it was introduced by AlexNet in a successful way, and it increases the dataset artificially by using different transformations, while maintaining the same label.

Despite the power of the CNN, as they work within supervised learning it means a disadvantage. To get a right learning require a large amount of data labeled by humans, which can be difficult to achieve in a real-world application. The more input data in the network, the better the output will be.

We know that CNN are the state-of-the-art of classification task due to the superior precision they offer over other methods of machine learning. However, these models require around 10^9 or 10^{10} labeled data to achieve this good performance [14]. There are several datasets to train the systems, but when the application is delicate or too new, the existing datasets are useless and creating a new specialized dataset for our problem can be too expensive and we must take into account that it is not fast.

Therefore, transfer learning is a good option when the size of the dataset is very small. It helps not to have to train from scratch, which increases the chances of obtaining better results.

As a further research, we would like to analyze more datasets to detect not only colors but also borders, tiny objects and complex structures. We also propose to create our own DNN from scratch by using Keras and its parallel mode, so we can obtain similar results to IBM Cloud, GoogLeNet and AlexNet.

References

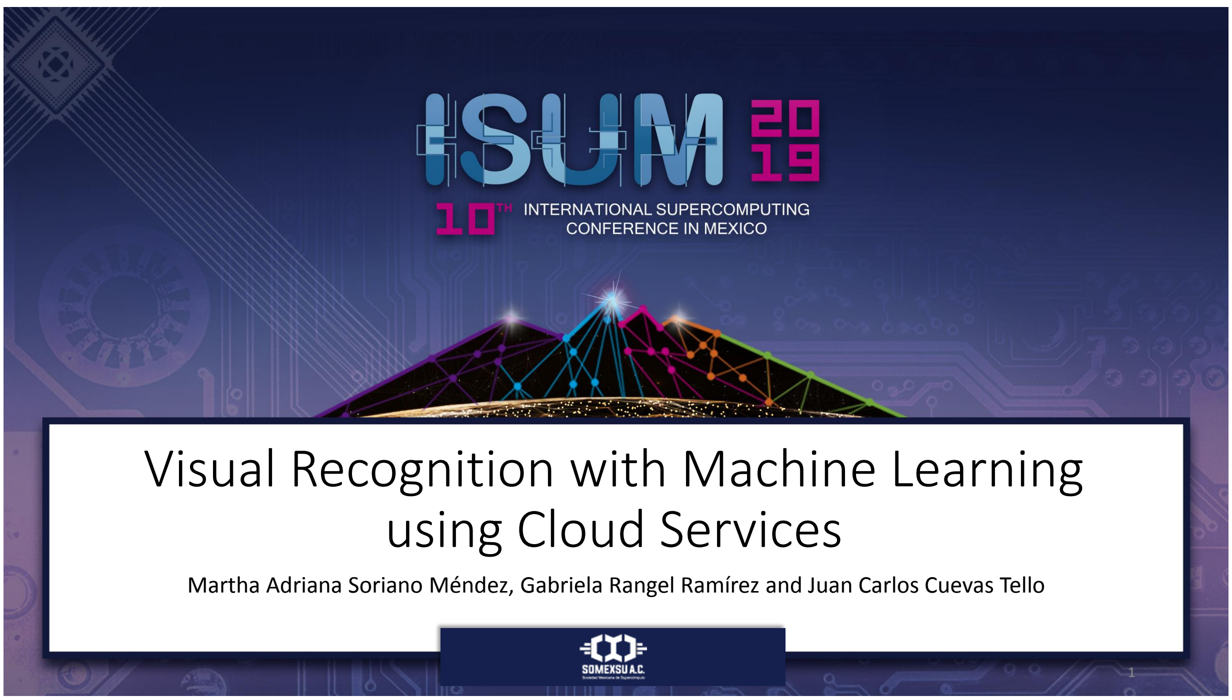
1. Nvidia GK110 [GeForce GTX 780].
2. Mário Campos, Teresa Martins, Manuel Ferreira, and Cristina Santos. Detection of defects in automotive metal components through computer vision. In *Industrial*

Electronics, 2008. ISIE 2008. IEEE International Symposium on, pages 860–865. IEEE, 2008.

3. Yangqing Jia Christian Szegedy, Wei Liu. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.
4. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
5. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT Press, 2016.
6. IBM Academic Initiative-https://developer.ibm.com/academic/.
7. Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.
8. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.
9. A. Luckow, M. Cook, N. Aschcraft, E. Weill, E. Djerekarov, and B. Vorster. Deep learning the automotive industry: Applications and tools. In *IEEE-2016*, 2016.
10. A. Luckow, K. Kennedy, M. Ziolkowski, E. Djerekarov, M. Cook, E. Duffy, M. Schleiss, B. Vorster, E. Weill, and A. Kulshrestha. Applications, architectures and best practices for AI vision systems in automotive manufacturing. In *ICSE-2018*, 2018.
11. MathWorks. Introducing deep learning with MATLAB. Technical report, The MathWorks, Inc., 2017.
12. Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.
13. T.M. Mitchell. *Machine Learning*. Mc Graw-Hill, 1997.
14. Information Innovation Office. Broad agency announcement learning with less labels (lwll). Technical report, Defense Advanced Research Projects Agency, 08 2018.
15. Mohan Karthik Ramaraj. A training assistant tool for the automated visual inspection system. Master's thesis, Clemson University, 2015.
16. A. Tank. This is the year of the machine learning revolution. *Entrepreneur*, 2017. https://www.entrepreneur.com/article/287324.
17. Michael Wittmann. *World Robotics 2017 Industrial Robots*. 2017.

Chapter 4

Presentation



Industry 4.0

- Rapid increase in productivity and significant energy and material savings.
- The automotive industry faces challenges such as regionalization, saturation, globalization, technological advances, new competitors and its continuous restructuring.
- The benefits are the improvement of quality processes, standardization in manufacturing, reduction of material waste, cost reduction, increased production and security.

Intelligent Factory

2

Problem

BMW IT Research Center has been working in three AI applications:

- *Visual inspection of vehicles and parts* (Visual Inspection)
- *Automated detection of bar code labels on boxes* to optimize the goods received process (Deep Logistics)
- *Detection of the trailer ID on trucks in trailer yard* (Deep Yard Management).

3

Problem <2>

Clemson University with BMW IT Research Center:

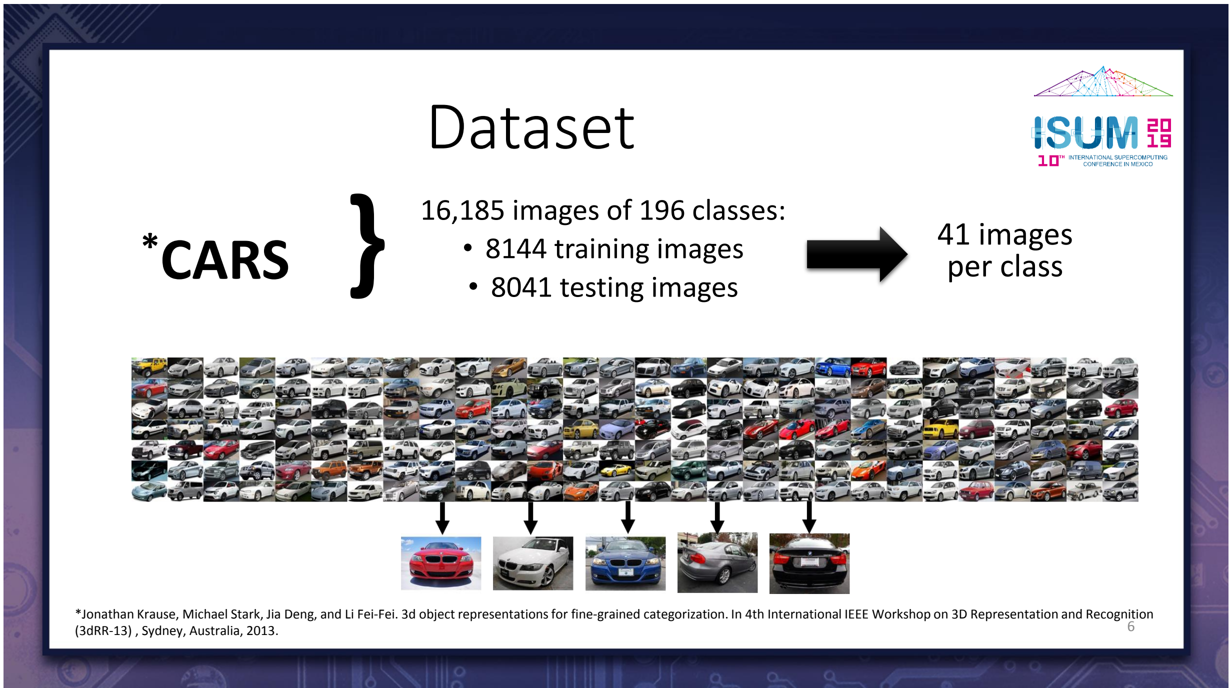
- Assist the human inspector checking whether a part has been installed or not.
- Need to classify large number of images collected from hundreds of cameras in the plant in a central location.

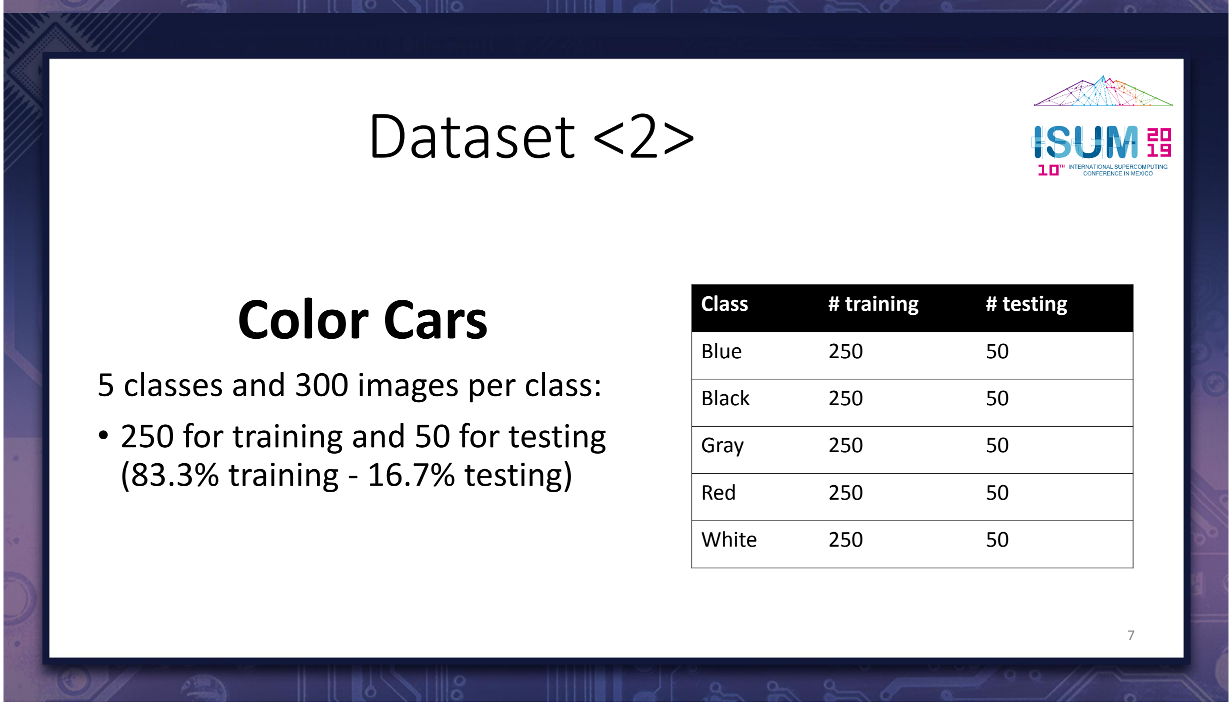
4

Objective

To measure of the performance from different machine learning algorithms for visual recognition through three different technologies: MATLAB, IBM Cloud - Visual Recognition and Keras library.

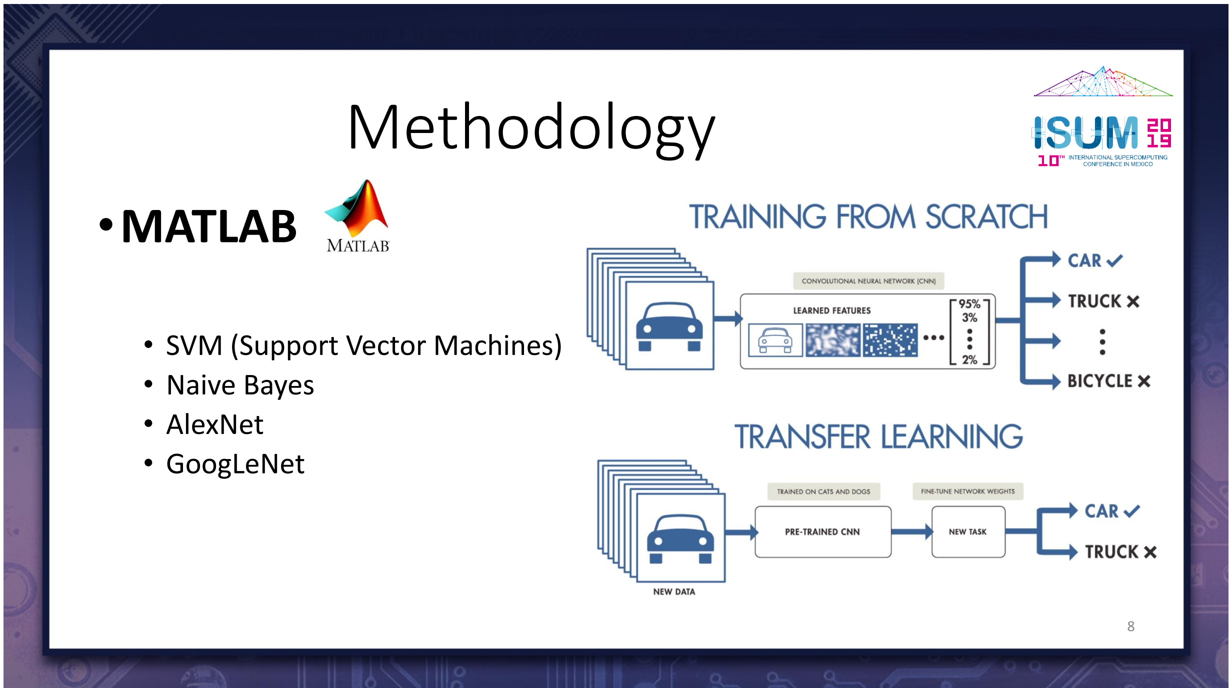
5

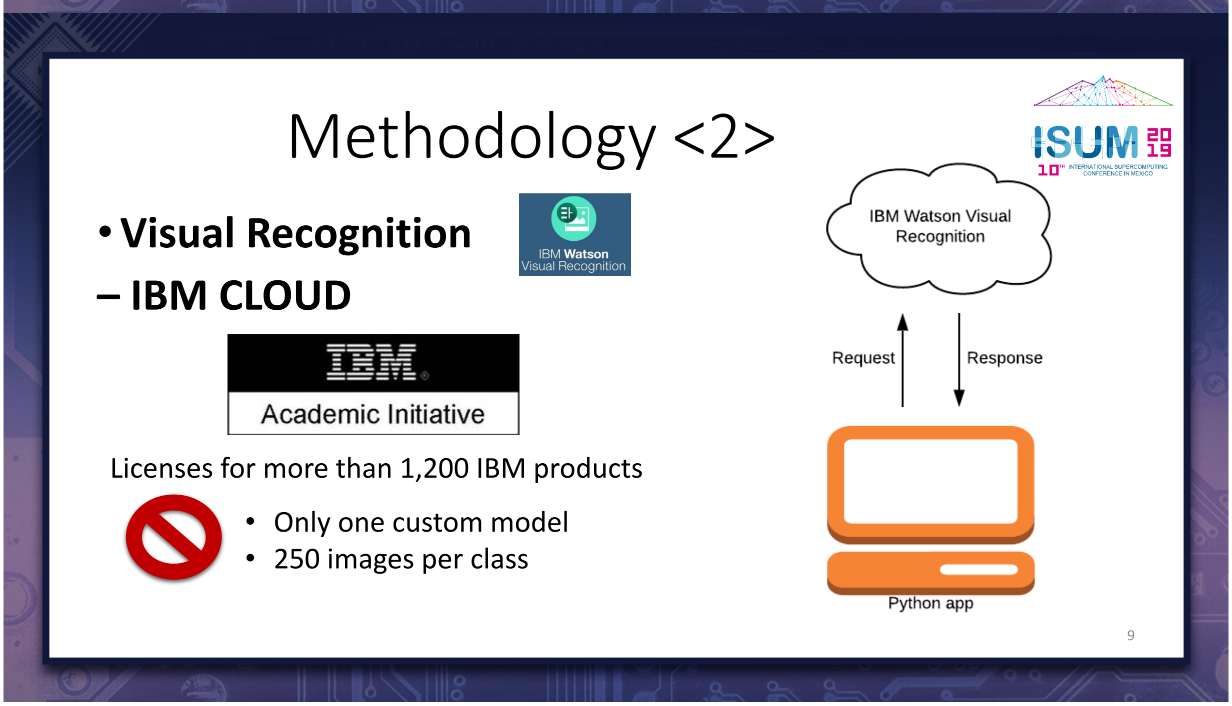

Dataset
ISUM 2019
10th INTERNATIONAL SUPERCOMPUTING CONFERENCE IN MEXICO
*CARS
16,185 images of 196 classes:
• 8144 training images
• 8041 testing images
41 images per class
*Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13) , Sydney, Australia, 2013.
6

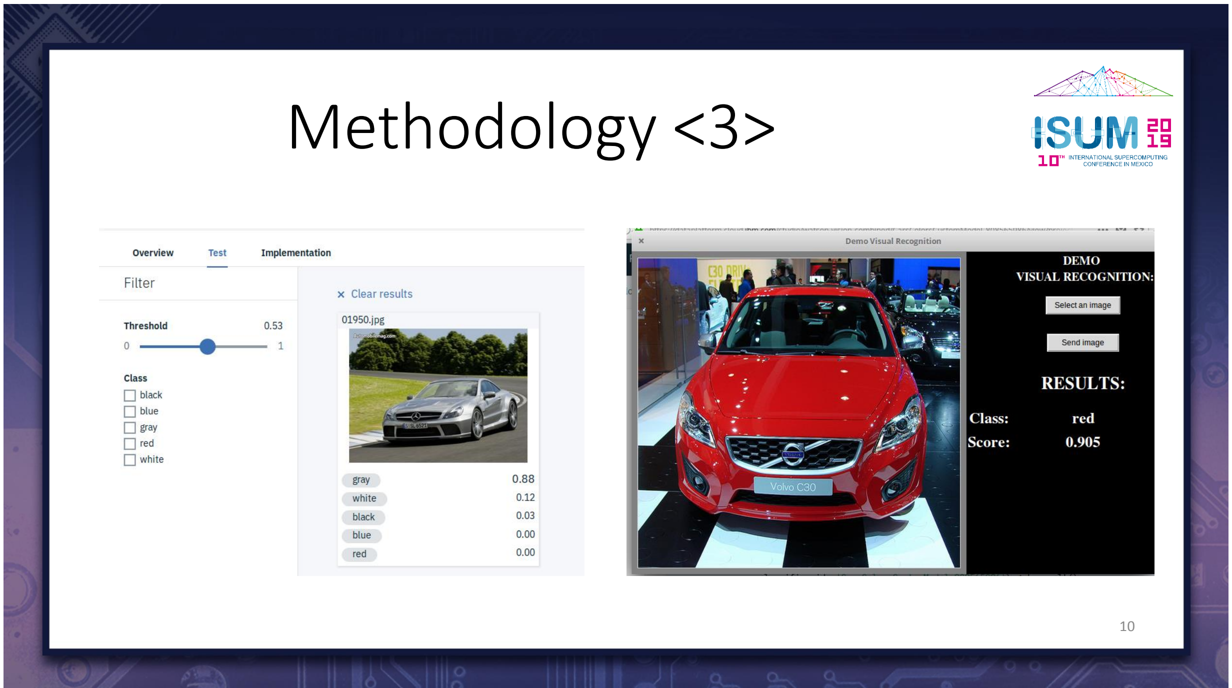

Dataset <2>
ISUM 2019
10th INTERNATIONAL SUPERCOMPUTING CONFERENCE IN MEXICO
Color Cars
5 classes and 300 images per class:
• 250 for training and 50 for testing (83.3% training - 16.7% testing)

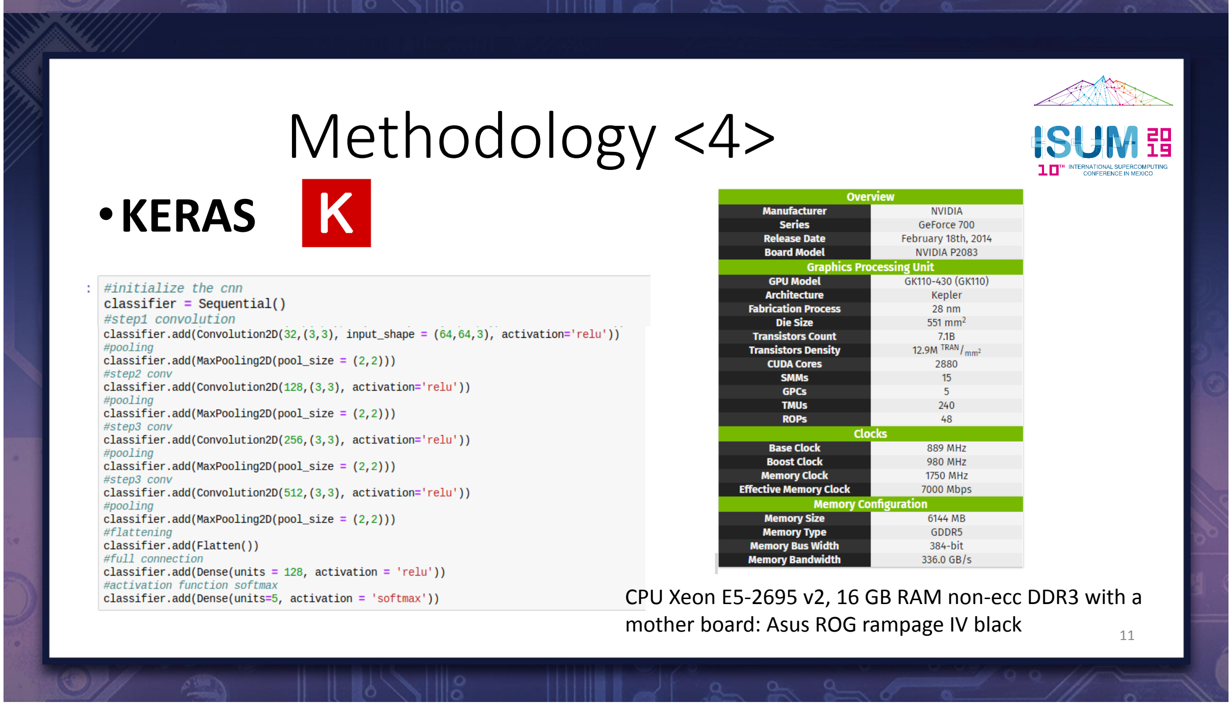
Class	# training	# testing
Blue	250	50
Black	250	50
Gray	250	50
Red	250	50
White	250	50

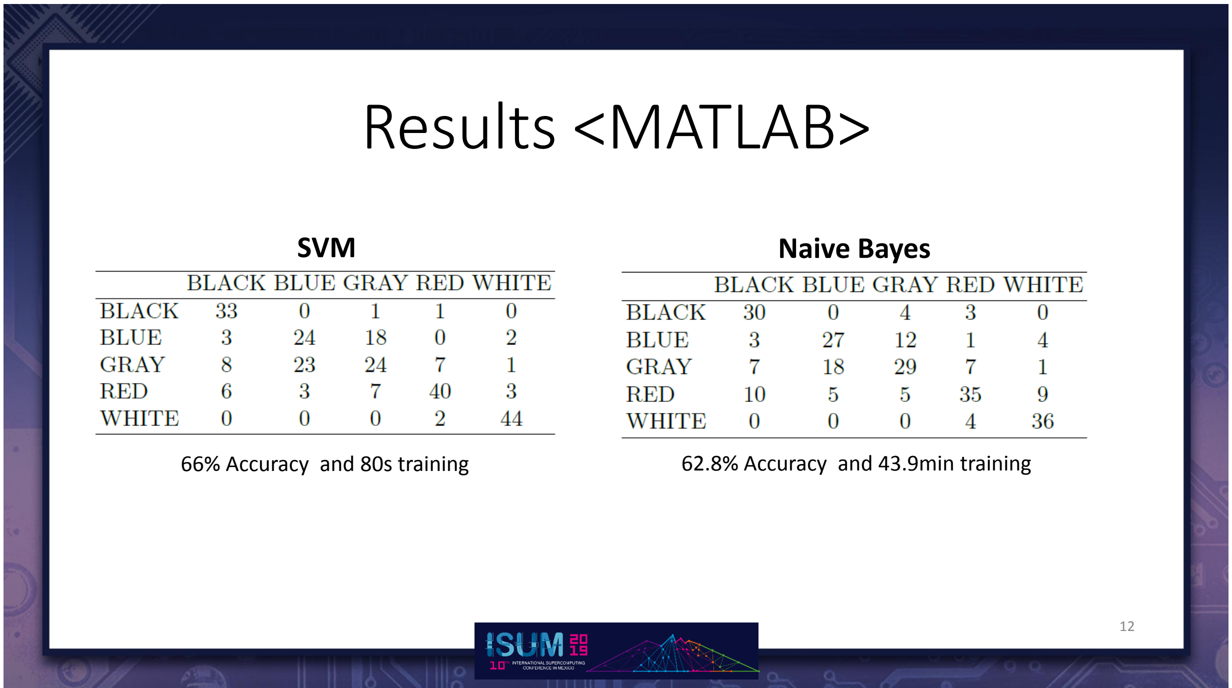
7


Methodology
ISUM 2019
• MATLAB
MATLAB
• SVM (Support Vector Machines)
• Naive Bayes
• AlexNet
• GoogLeNet
TRAINING FROM SCRATCH
CONVOLUTIONAL NEURAL NETWORK (CNN)
LEARNED FEATURES
95%
3%
2%
CAR ✓
TRUCK ✗
BICYCLE ✗
TRANSFER LEARNING
TRAINED ON CATS AND DOGS
FINE-TUNE NETWORK WEIGHTS
PRE-TRAINED CNN
NEW TASK
NEW DATA
CAR ✓
TRUCK ✗
8


Methodology <2>
ISUM 2019
• Visual Recognition
– IBM CLOUD
IBM Watson Visual Recognition
IBM
Academic Initiative
Licenses for more than 1,200 IBM products
• Only one custom model
• 250 images per class
IBM Watson Visual Recognition
Request
Response
Python app
9


Methodology <3>
Overview
Test
Implementation
Filter
Threshold
0.53
0
1
Class
black
blue
gray
red
white
× Clear results
01950.jpg
gray 0.88
white 0.12
black 0.03
blue 0.00
red 0.00
Demo Visual Recognition
DEMO
VISUAL RECOGNITION:
Select an image
Send image
RESULTS:
Class: red
Score: 0.905
10


Methodology <4>
• KERAS
#initialize the cnn
classifier = Sequential()
#step1 convolution
classifier.add(Convolution2D(32,(3,3), input_shape = (64,64,3), activation='relu'))
#pooling
classifier.add(MaxPooling2D(pool_size = (2,2)))
#step2 conv
classifier.add(Convolution2D(128,(3,3), activation='relu'))
#pooling
classifier.add(MaxPooling2D(pool_size = (2,2)))
#step3 conv
classifier.add(Convolution2D(256,(3,3), activation='relu'))
#pooling
classifier.add(MaxPooling2D(pool_size = (2,2)))
#step3 conv
classifier.add(Convolution2D(512,(3,3), activation='relu'))
#pooling
classifier.add(MaxPooling2D(pool_size = (2,2)))
#flattening
classifier.add(Flatten())
#full connection
classifier.add(Dense(units = 128, activation = 'relu'))
#activation function softmax
classifier.add(Dense(units=5, activation = 'softmax'))
Overview
Manufacturer NVIDIA
Series GeForce 700
Release Date February 18th, 2014
Board Model NVIDIA P2083
Graphics Processing Unit
GPU Model GK110-430 (GK110)
Architecture Kepler
Fabrication Process 28 nm
Die Size 551 mm2
Transistors Count 7.1B
Transistors Density 12.9M TRAN/mm2
CUDA Cores 2880
SMMs 15
GPCs 5
TMUs 240
ROPs 48
Clocks
Base Clock 889 MHz
Boost Clock 980 MHz
Memory Clock 1750 MHz
Effective Memory Clock 7000 Mbps
Memory Configuration
Memory Size 6144 MB
Memory Type GDDR5
Memory Bus Width 384-bit
Memory Bandwidth 336.0 GB/s
CPU Xeon E5-2695 v2, 16 GB RAM non-ecc DDR3 with a mother board: Asus ROG rampage IV black
11



Results <MATLAB>

SVM

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	33	0	1	1	0
BLUE	3	24	18	0	2
GRAY	8	23	24	7	1
RED	6	3	7	40	3
WHITE	0	0	0	2	44

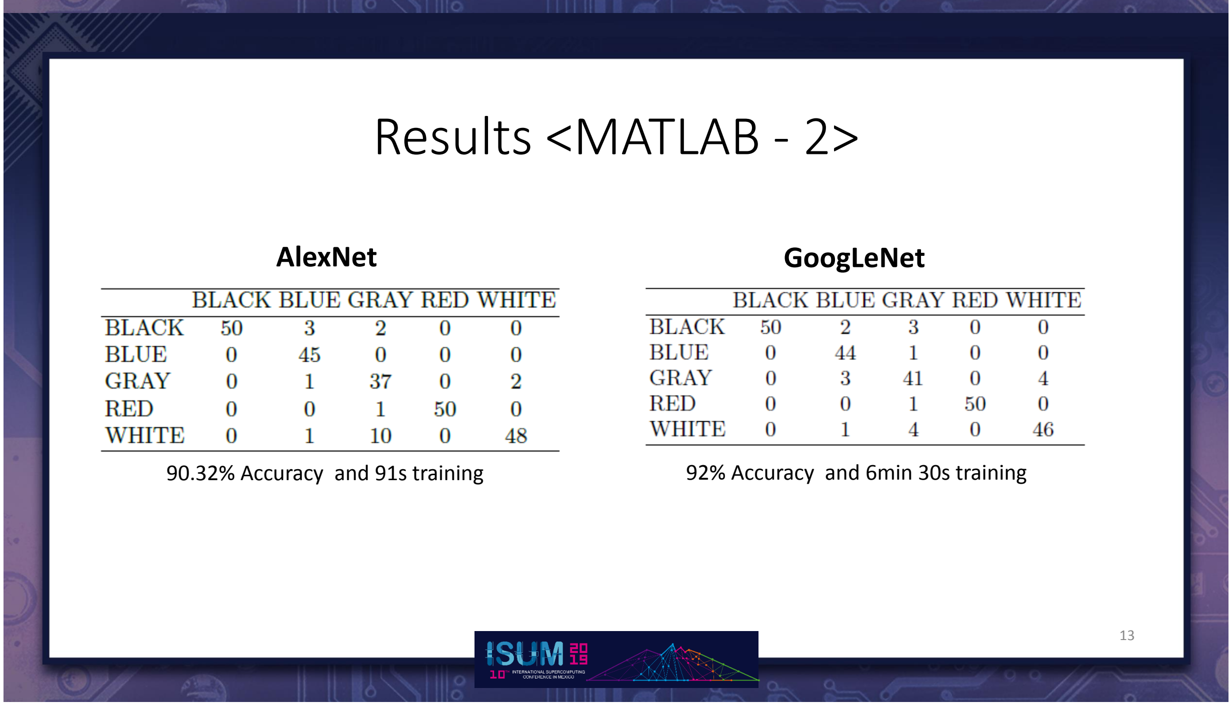
66% Accuracy and 80s training

Naive Bayes

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	30	0	4	3	0
BLUE	3	27	12	1	4
GRAY	7	18	29	7	1
RED	10	5	5	35	9
WHITE	0	0	0	4	36

62.8% Accuracy and 43.9min training

12



Results <MATLAB - 2>

AlexNet

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	50	3	2	0	0
BLUE	0	45	0	0	0
GRAY	0	1	37	0	2
RED	0	0	1	50	0
WHITE	0	1	10	0	48

90.32% Accuracy and 91s training

GoogLeNet

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	50	2	3	0	0
BLUE	0	44	1	0	0
GRAY	0	3	41	0	4
RED	0	0	1	50	0
WHITE	0	1	4	0	46

92% Accuracy and 6min 30s training

13

Results <Visual Recognition – IBM Cloud>

Training time: 57s
Testing time: 237 seconds equivalent to 3.95 minutes.

Threshold 0.9

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	48	1	0	0	0	1
BLUE	0	39	0	0	1	10
GRAY	1	0	14	0	2	33
RED	0	0	0	50	0	0
WHITE	0	0	0	0	48	2

79.6% Accuracy

Threshold 0.8

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	48	1	0	0	0	1
BLUE	0	43	1	0	1	5
GRAY	2	1	26	0	7	14
RED	0	0	0	50	0	0
WHITE	0	0	0	0	50	0

86.8% Accuracy

14

Results <Visual Recognition – IBM Cloud>

Training time: 57s
Testing time: 237 seconds equivalent to 3.95 minutes.

Threshold 0.6

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	49	1	0	0	0	0
BLUE	1	45	2	0	2	0
GRAY	4	1	33	0	13	-1
RED	0	0	0	50	0	0
WHITE	0	0	0	0	50	0

90.8% Accuracy

Threshold 0.5

	BLACK	BLUE	GRAY	RED	WHITE	LOST
BLACK	49	1	0	0	0	0
BLUE	2	45	2	0	2	-1
GRAY	5	1	37	0	14	-7
RED	0	0	0	50	0	0
WHITE	0	0	0	0	50	0

92.4% Accuracy

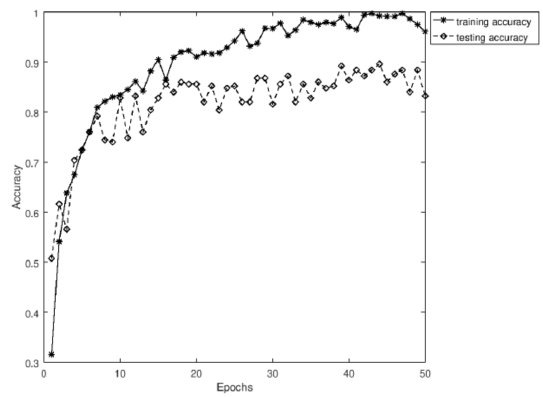
15

Results <Keras – Sequential>

Training time: 2879.9627s, equivalent to 47.99min

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	35	5	5	2	4
BLUE	2	45	2	0	2
GRAY	5	1	37	0	14
RED	0	0	0	50	0
WHITE	0	0	0	0	50

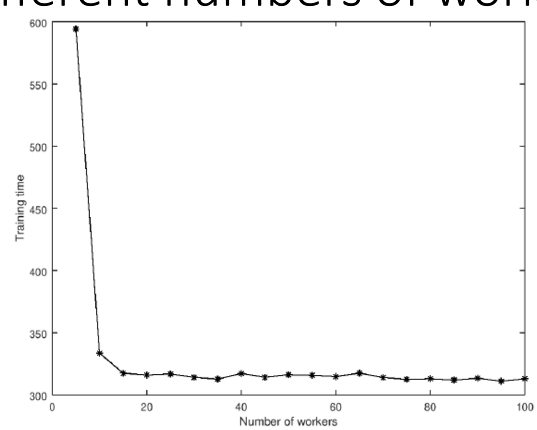
86.8% Accuracy



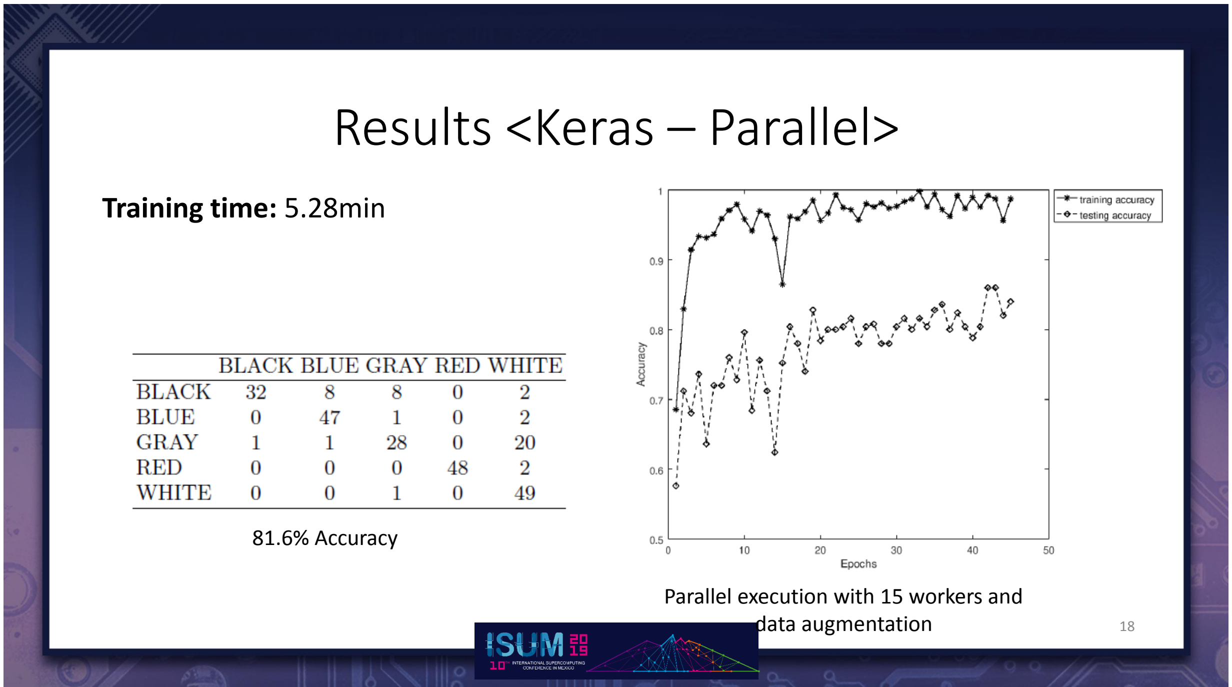
Sequential performance of CNN with Keras

16

Comparison of training time with different numbers of workers

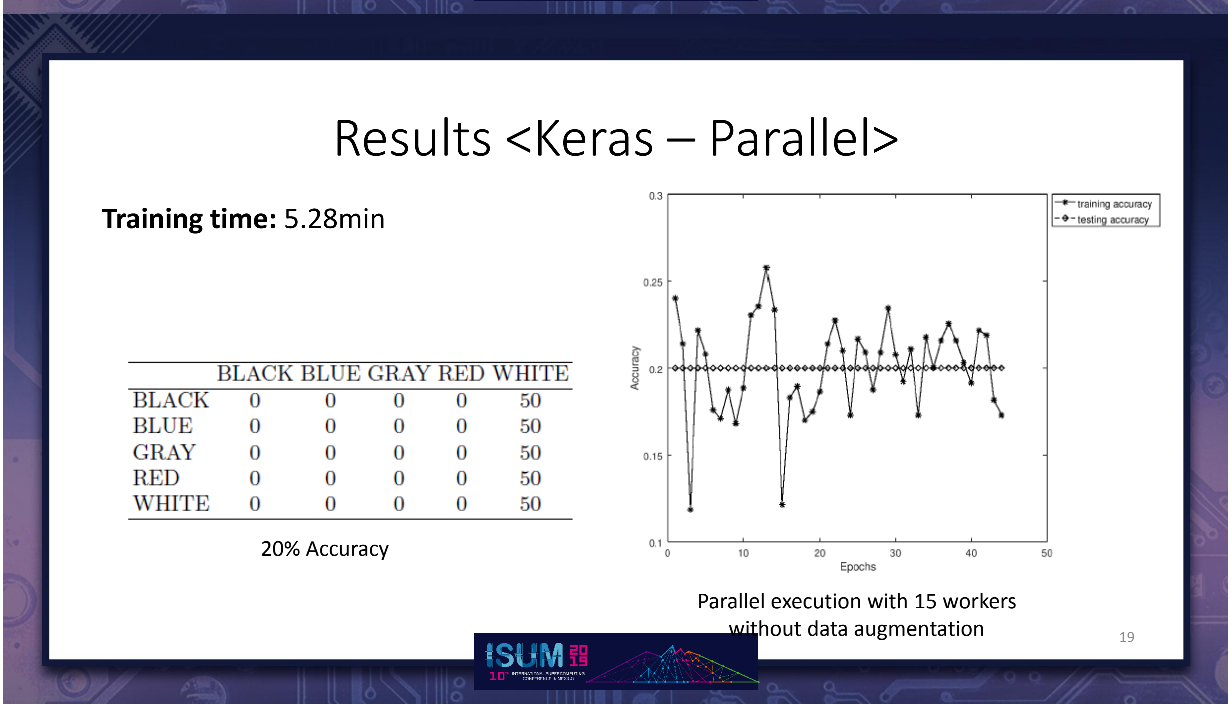


17


Results <Keras – Parallel>
Training time: 5.28min

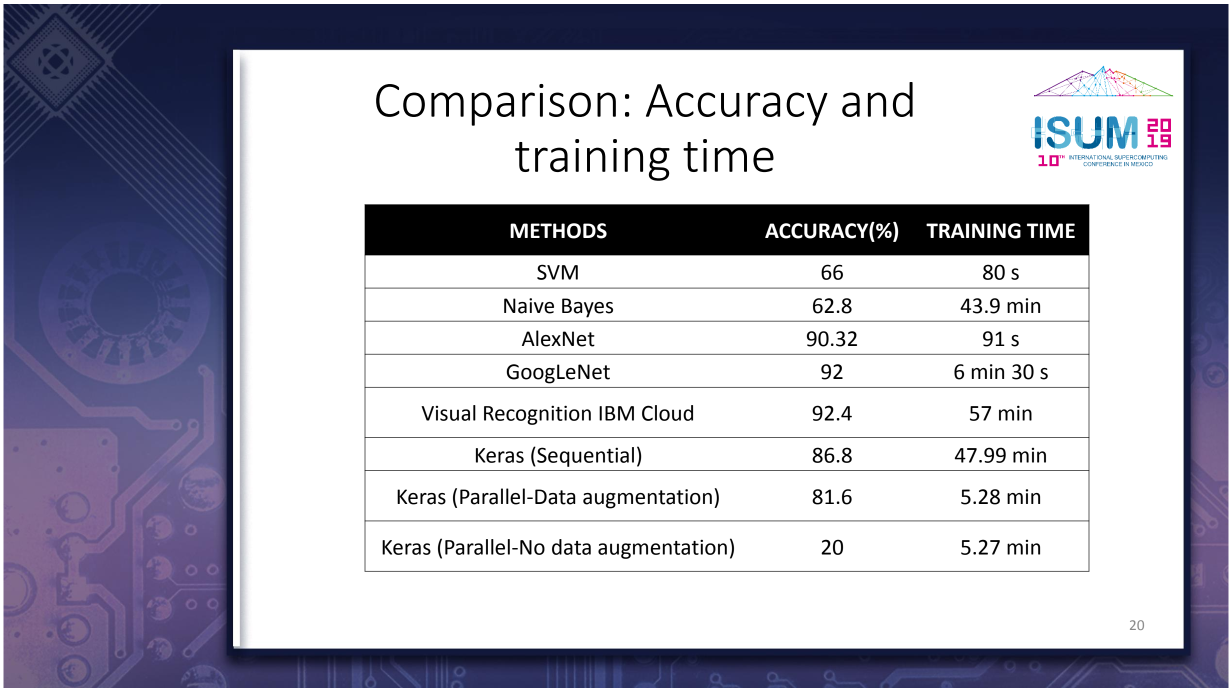
	BLACK	BLUE	GRAY	RED	WHITE
BLACK	32	8	8	0	2
BLUE	0	47	1	0	2
GRAY	1	1	28	0	20
RED	0	0	0	48	2
WHITE	0	0	1	0	49

81.6% Accuracy
training accuracy
testing accuracy
Accuracy
Epochs
Parallel execution with 15 workers and data augmentation
18


Results <Keras – Parallel>
Training time: 5.28min

	BLACK	BLUE	GRAY	RED	WHITE
BLACK	0	0	0	0	50
BLUE	0	0	0	0	50
GRAY	0	0	0	0	50
RED	0	0	0	0	50
WHITE	0	0	0	0	50

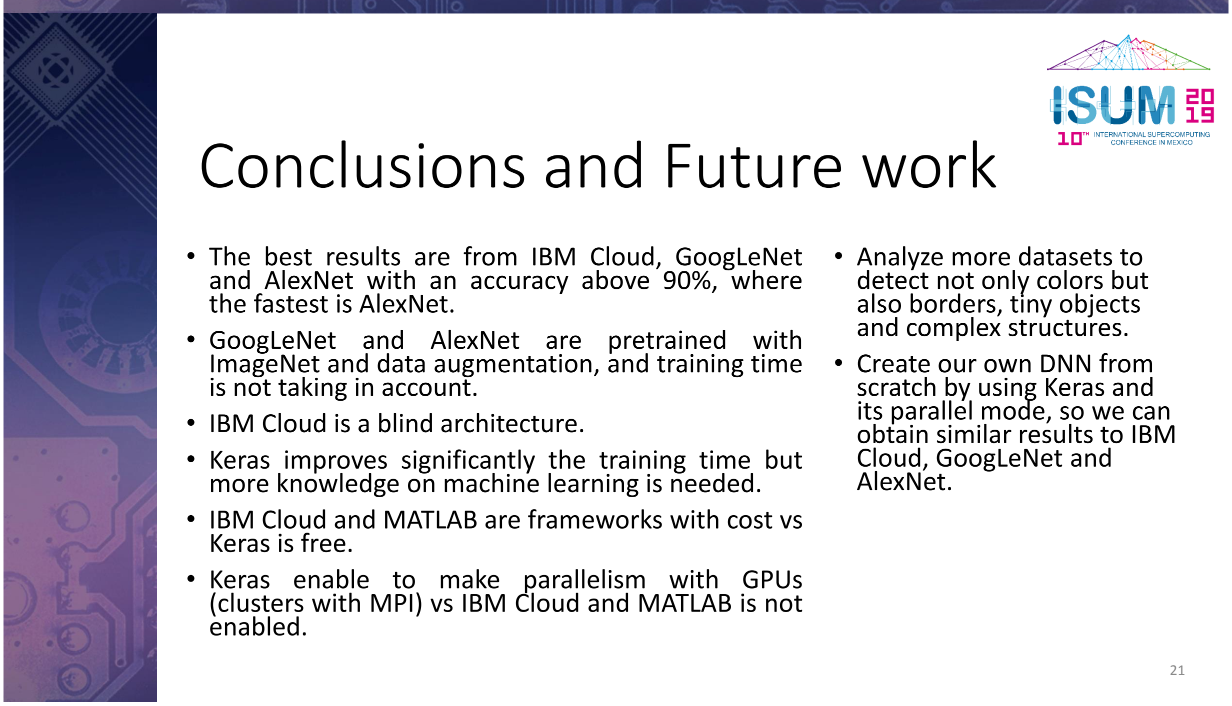
20% Accuracy
training accuracy
testing accuracy
Accuracy
Epochs
Parallel execution with 15 workers without data augmentation
19



Comparison: Accuracy and training time

METHODS	ACCURACY(%)	TRAINING TIME
SVM	66	80 s
Naive Bayes	62.8	43.9 min
AlexNet	90.32	91 s
GoogLeNet	92	6 min 30 s
Visual Recognition IBM Cloud	92.4	57 min
Keras (Sequential)	86.8	47.99 min
Keras (Parallel-Data augmentation)	81.6	5.28 min
Keras (Parallel-No data augmentation)	20	5.27 min

20



Conclusions and Future work

- The best results are from IBM Cloud, GoogLeNet and AlexNet with an accuracy above 90%, where the fastest is AlexNet.
- GoogLeNet and AlexNet are pretrained with ImageNet and data augmentation, and training time is not taking in account.
- IBM Cloud is a blind architecture.
- Keras improves significantly the training time but more knowledge on machine learning is needed.
- IBM Cloud and MATLAB are frameworks with cost vs Keras is free.
- Keras enable to make parallelism with GPUs (clusters with MPI) vs IBM Cloud and MATLAB is not enabled.
- Analyze more datasets to detect not only colors but also borders, tiny objects and complex structures.
- Create our own DNN from scratch by using Keras and its parallel mode, so we can obtain similar results to IBM Cloud, GoogLeNet and AlexNet.

21


THANK YOU
ISUM 2019
10th INTERNATIONAL SUPERCOMPUTING CONFERENCE IN MEXICO

Appendix A

Theorical Framework

A.1 Artificial intelligence

Artificial Intelligence (AI) is one of the areas of great interest for companies. It can be defined as the ability of a computer program or machine to learn and think. AI is a branch of computer science which aims to understand the essence of intelligence and reproduce it on a machine which responds similarly to human intelligence.

Artificial Intelligence, officially emerges from the works published in the 1940s that did not have great importance or impact. But from the significant work in 1950 by Alan Turing, an English mathematician, a new discipline of information science emerge. However, the term was first coined until 1956 by John McCarthy, Marvin Minsky and Claude Shannon during Dartmouth conference as “the science and ingenuity of making intelligent machines, especially intelligent calculation programs”.

Some AI problems that were proposed at this conference were:

- Automatic Computers
- How can a computer be programmed to use language?
- Neural Networks
- Theory of the calculation dimension

- Self-improvement
- Abstractions
- Randomness and Creativity

These scientists pointed out that society would be surrounded by intelligent machines in less than ten years. The truth is that it was not so, and that is, this technology developed timidly until the 1990s, when the golden age of AI really begins.

As of 1990, most of the big technology companies start making huge investments in this area. Due they realized that needed to improve the processing and analysis capacity of the enormous amount of data that was coming. Research in this area includes machine learning, natural language processing, expert systems, vision, speech recognition, planning and robotics.

A.2 Machine learning

Machine learning (ML) area is a computer science field that is responsible for learning given a set of data. In other words, it is responsible for representing the structure and generalizing behaviors of the given data. It generally focuses on analyzing data for patterns and relationships.

A computer program learns from experience E with respect to a task T and some measure of performance P. The above is true if the performance in task T, measured by P, improves with experience E. [6].

ML is the field where the most important advances of the AI are being carried out. In practical terms, machine learning is the science that is responsible for making computers perform actions without the need for explicit programming. The main idea here is you can provide data to the machine learning algorithms and then use them to know how to make predictions or guide decisions [4].

Some examples of machine learning algorithms include the following: decision trees, clustering algorithms, genetic algorithms, Bayesian networks and deep learning.

Using machine learning can improve some automotive fields like advanced driving assistance systems, autonomous driving and others fields related with sales and after sales processes, creating applications with a high potential. Machine learning is used for process all the information obtained from sensors, cameras, etc., and it has been proven as a capable tool for detecting patterns in data and deriving predictions.

There are two types of machine learning:

- Supervised: This branch of the ML takes care of the problems that bring data with labels. These types of algorithms seek to generalize and predict based on the information provided.

- Not supervised: In the absence of a more creative name, this branch of ML takes care of the rest of the problems, that is, those that do not bring the data with labels. For example, given a group of users of a social network, group them (clustering) into communities. These types of algorithms seek to extract structure and find patterns from the information provided.

A.3 Deep learning

Deep learning (DL) refers to a set of machine learning algorithms that utilize large neural networks with many hidden layers for tasks, such as image classification, speech recognition and language understanding as examples [3]. Deep learning can help to organize data and improve data collection process from visual inspection in manufacturing process. In social media analytics, deep learning can assist and improve data collection and analysis. In case of robots and smart machines, enables self-learning robots to become more intelligent over their lifetime.

In figure A.1 a deep learning model is shown, showing the layers that are used in this process.

The first layer is called the visible layer or input layer because it contains the values that are possible to observe, in this case the image of a person is shown as a collection of pixel values, which is not easy to understand for a computer. The hidden

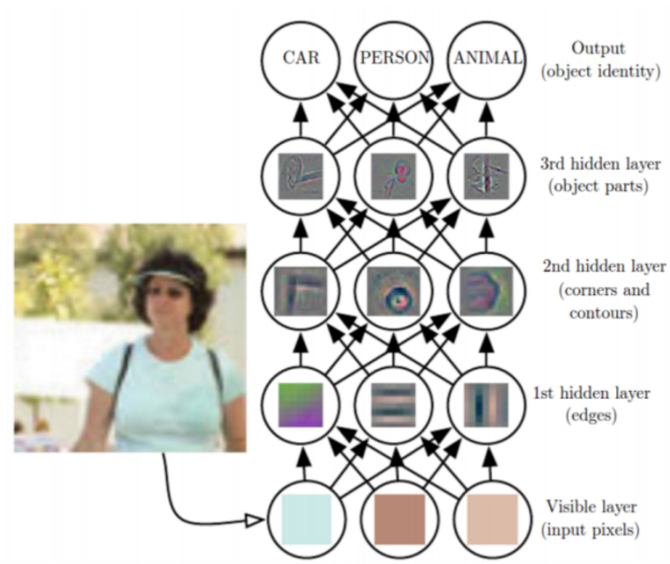


Figure A.1: Deep learning layers [2]

layers are responsible for extracting increasingly abstract characteristics. They are named in that way because their values are not given in the data, therefore the model must determine which concepts are useful to explain the relationships between the observed data. Given the pixels at the entrance, the first hidden layer can easily identify the edges by comparing the brightness of the neighboring pixels, then the second hidden layer identifies corners and contours that are collections of borders. The third hidden layer identifies parts integers of specific objects finding collections of contours and corners. Finally the image is classified in the category that belongs to.

The function of mapping from a set of pixels to an object identity is complicated. Learning or evaluating this mapping seems insurmountable whether addressed directly. Deep learning solves this difficulty by dividing complicated mapping into a series of simple nested mappings, each described by a different layer. [2].

One of the most common vision problem solved by deep learning technique is image classification. In this kind of problem the algorithm try to assign an input image one label from a fixed set of categories. The image classification process consists of the following three steps:

- Input: Set of N images labeled with K different classes (training set).

- Learning: Consists of taking the training set to learn each one of the different classes, it means learning a model or training a classifier.

- Evaluation: It consists in evaluating the quality of the trained classifier with a new set of images, so that it predicts the class to which a test image belongs. In addition, the actual labels of those images are compared with those predicted by the classifier.

For the computer an image is represented as a three-dimensional array of numbers. For example, the cat image A.3 is 248 pixels wide, 400 pixels high and has 3 color channels, red, green and blue (RGB). Therefore, the image consists of 248 x 400 x 3 numbers, or a total of 297,600 numbers. Each number is an integer that ranges from 0 (black) to 255 (white). Our task is to turn this quarter of a million numbers into a single label, such as "cat".

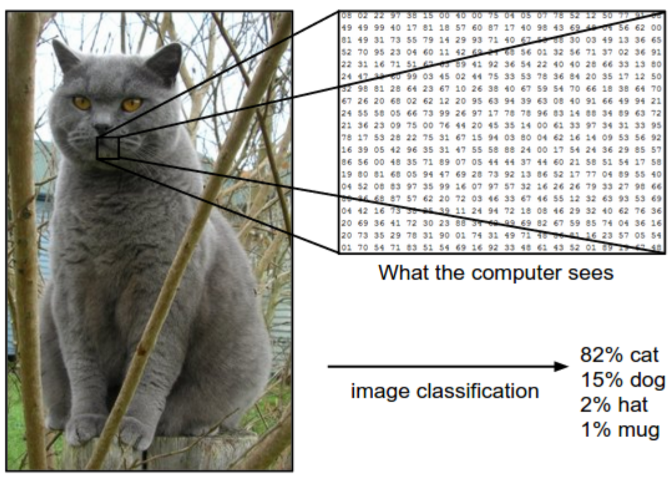


Figure A.2: Cat image classification [1]

It is trivial for a person to classify an object, for a computer vision algorithm it is more complicated, therefore it presents several challenges, among them are:

- Viewpoint variation. A single instance of an object can be oriented in many ways with respect to the camera.

- Scale variation. Visual classes often exhibit variation in their size (size in the real world, not only in terms of their extent in the image).

- Deformation. Many objects of interest are not rigid bodies and can be deformed in extreme ways.
- Occlusion. The objects of interest can be occluded. Sometimes only a small portion of an object (as little as few pixels) could be visible.
- Illumination conditions. The effects of illumination are drastic on the pixel level.
- Background clutter. The objects of interest may blend into their environment, making them hard to identify.
- Intra-class variation. The classes of interest can often be relatively broad, such as chair. There are many different types of these objects, each with their own appearance.

A.4 Neural network

A neural network is a representation inspired by the functioning of the human brain. It is formed by processing units called neurons and they have connections between them. They are also organized in the following layers:

- Input layer: Data is entered depending on the task to be performed.
- Hidden layers: Data processing is carried out and is positioned between the input layer and the output layer.
- Output layer: It is responsible for representing the values of the class to which the image belongs in classification tasks.

Neural networks are modeled as neurons collections contained in an acyclic graph, it means outputs of some neurons become inputs of others. These networks are organized into different layers of neurons, the most common is the completely connected, in which the neurons between two adjacent layers are connected and the neurons that belong to a layer do not share connections. It should be noted that exists many architectures to represent neural networks like perceptron, Boltzmann machine, Kohonen network, Hopfield network and so on. The next picture represents a basic Convolutional Neural Network. This network was used in Keras framework evaluation.

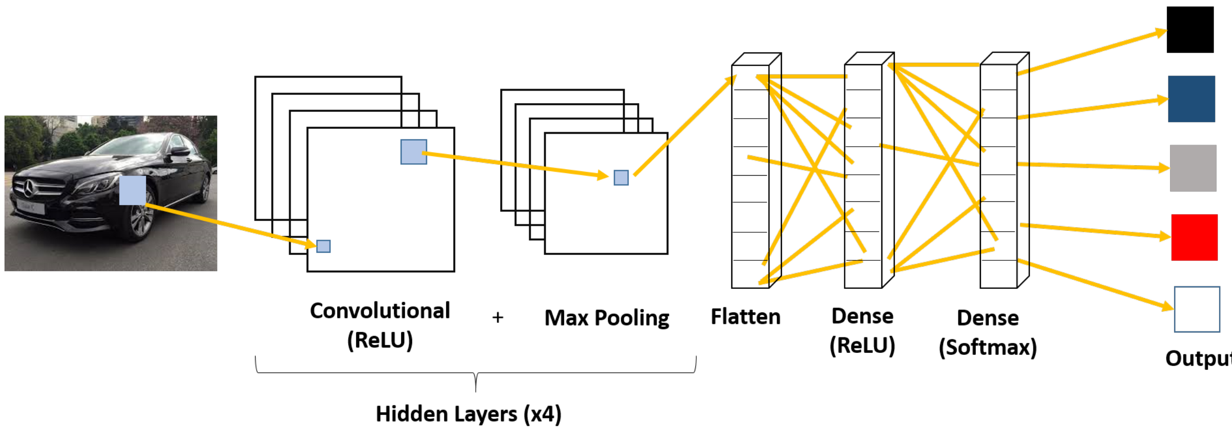


Figure A.3: CNN for Keras framework [1]

Some applications of neural networks are [9]:

- Classification of email.
- Speech recognition.
- Character recognition.
- Face Recognition

A.4.1 Data augmentation

Deep networks need a large amount of data to get good results, if you have a small set of images, data augmentation is a good technique to achieve good performance in training images. There are several ways to increase the data and among them are different transformations like:

- Flip
- Crop

- Color fluctuation
- Scaling
- Rotation
- Translation
- Masks

Figure A.4: Example of data augmentation [8]

A.4.2 Network initialization

Before the network is trained, the parameters have to be initialized and there are different ways to do it:

- Zero initialization: Assumes with data normalization approximately half of the weights will be positive and the other will be negative.
- Initialization with small random numbers: Technique in which weights are initialized with small random numbers, close to zero. It is known as symmetry breaking.

A.5 Convolutional Neural Network

In recent years, Deep learning models have exploited multiple layers of non-linear information processing, for feature extraction and transformation, as well as pattern analysis and classification, have been shown to overcome all these challenges. Among them, CNNs have become the leading architecture for most image recognition, classification and detection tasks ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a benchmarking in classification of object categories and detection of hundreds of objects and millions of images. The challenge has been made annually since 2010, attracting the participation of more than 50 institutions. It consists of two components:

- Data set available to the public
- Annual competition

This data allows development and comparison of object category recognition algorithms and competition provides a way to track progress and discuss the lessons learned from the most successful and innovators algorithms every year. This data set contains a training set of hand-tagged images. A set of test images with manual annotations retained. Participants train their algorithms using the training set and then automatically label the test images. Predicted tagging is sent to the evaluation server [5]. Deep convolutional neural networks have dominated classification tasks. In fact, they have won at each event since 2012 A.5

ConvNet (Convolutional Network) or CNN (Convolutional Neural Networks) are feedforward networks, in which information flows in only one direction, from inputs to outputs. The CNN architecture has many variants but in general, they consist of convolutional layers and pooling, which are grouped into modules. Followed by one or more fully connected layers, as in common feedforward neural networks A.6. The modules are stacked on top of each other to form a deep model [7]

Year	Team	Number of Layers	General Contribution	Position	References
2010	NEC	Shallow	Fast feature extraction, data compression, SVM classifier	First	Lin et al., 2011
2011	XRCE	Shallow	High-dimensional image signatures, data compression, SVM classifier	First	Perronnin et al., 2010; Sánchez & Perronnin, 2011
2012	SuperVision	8	Efficient GPU-based DCNN, with Dropout and several other innovations	First	Krizhevsky et al., 2012
2013	Clarifai	8	DCNN architecture based on deconvolutional visualization technique	First	Zeiler & Fergus, 2014; Zeiler et al., 2011
2014	GoogLeNet	22	DCNN architectural design based on Hebbian principle and multiscale ideas	First	Szegedy, Vanhoucke et al., 2015
2014	VGG	19	Improvements to DCNN convolutional layers, increased network depth	Second	Simonyan & Zisserman, 2014
2015	MSRA	152	Introduction of deep residual learning for ultra DCNNs	First	He et al., 2015b

Figure A.5: ImageNet results [7]

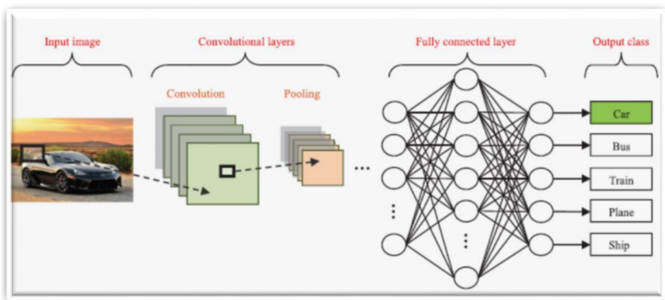


Figure A.6: Convolutional Neural Network - CNN [7]

A.5.1 Layers

Convolutional Layer

The convolution layer receives the image as input and then applies on it a filter or kernel that returns a map of the characteristics of the original image, in this way the size of the parameters is reduced. The convolution takes advantage of three ideas that help improve any machine learning system and are:

- Dispersed interactions: By applying a smaller filter on the original input, the amount of parameters and calculations is significantly reduced.

- Shared parameters: They are those that refer to sharing the parameters between the different types of filters, this helps the efficiency of the system.

- Equivalent representation: They indicate that if the inputs change then the outputs will change in a similar way.

On the other hand, convolution provides a means to work with variable size inputs.

Pooling layer

Pooling layer is located after the convolutional layer. Its objective is to reduce the spatial dimensions of the input volume for the next convolutional layer without affecting the depth. The operation that is performed in this layer is also called sampling reduction due to the reduction in size and the loss of information is favorable because of the decrease in size that leads to a lower overload in the calculation in the next layers, in addition to May decrease overfitting. A layer in a convolutional network consists of three stages, in the first stage it represents convolutions in parallel to produce a set of linear activations, in the second stage each linear activation is executed through a non-linear activation function, in the third stage a pooling function is used to modify the output of the layer. In all cases the pooling layer helps to make an approximate representation invariant to small translations of the entrance. Invariant to translations means that if the input moves a little, the values of the grouped outputs do not change.

Fully connected layer

It is used at the end of the convolutional and pooling layers, in this layer each pixel is considered as a separate neuron as in a neural network. This layer is responsible for classifying and will have a number of neurons that corresponds to the number of classes to predict.

Appendix B

Register to IBM Cloud and Promo Code

Go to the IBM Cloud https://console.bluemix.net/registration/ to create a free account and fill the required information Fig 1. The email address you submit will become your IBM ID check. You should receive your confirmation email in a few minutes. Click the provided link to confirm your registration.

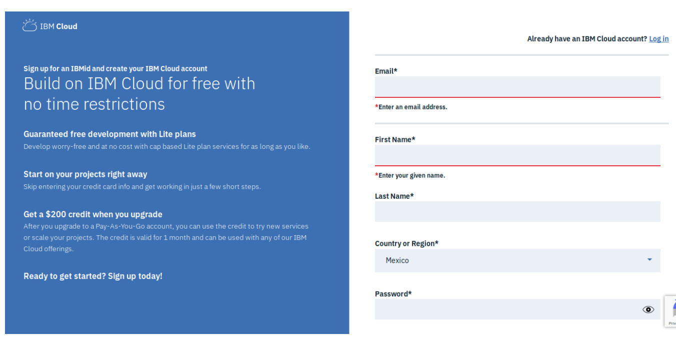


Figure B.1: Web site registration

Next, to obtain a free promo code from Academic Initiative IBM you should enter in https://ibm.onthehub.com, the same email you specified during registration for IBM Cloud.

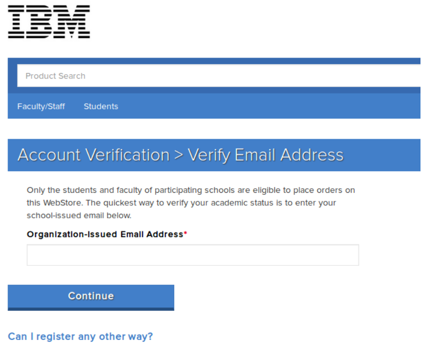


Figure B.2: Promo code

Once your account is verified, you could finish your registration filling the required information.

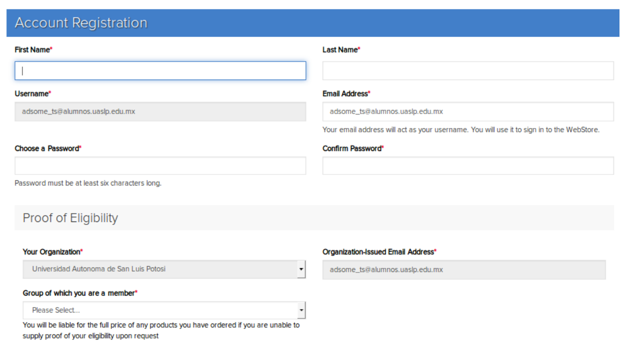


Figure B.3: Account verification

After that, you will have access to select the promo code. If you are a Teacher or Instructor you can request the promo code for 12 Months, if not, click on "¿Es usted elegible?" link to see the options you can get. These options will appear with a green check mark. In case you are a student you can apply for 6 Months promo code.

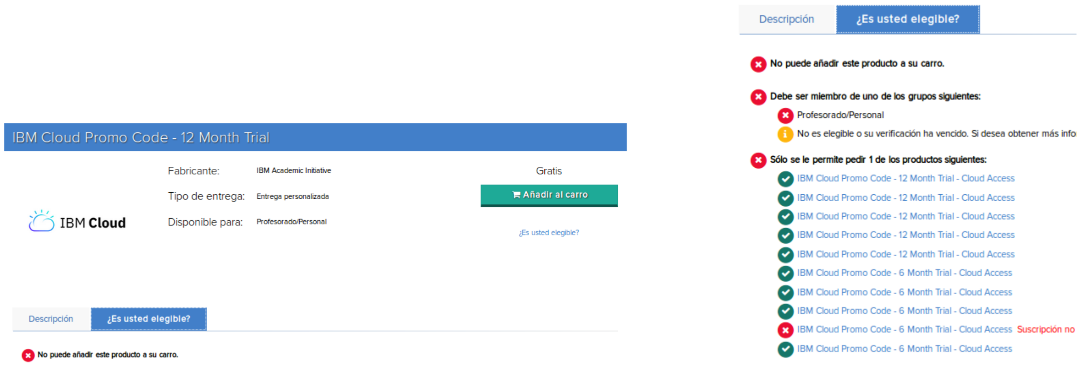


Figure B.4: Selecting promo code

Then you completed the previous steps, you will receive an email with your code. This code will be used later. As next step, log in IBM Cloud and go to Manage-Billing and usage-Billing. In Feature (Promo) Codes select Apply code and introduce the received code. Alternatively, you can apply the IBM Cloud Promo Code through: Manage → Account → Account Settings → Feature Codes → Apply Code.

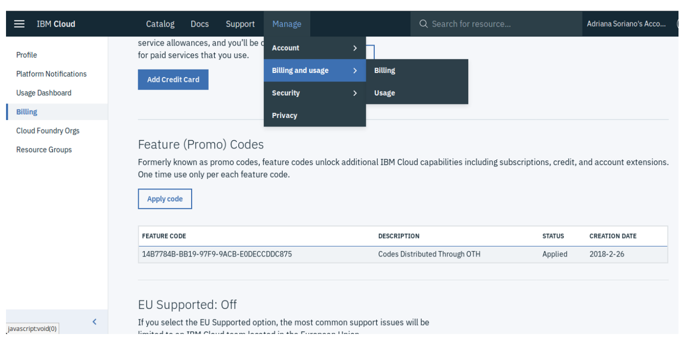


Figure B.5: Promo code activation

Once you have finished all the steps you can go to https://dataplatform.cloud.ibm.com/ and start using the services of IBM Cloud.

Appendix C

IBM Cloud: Watson Visual Recognition

As first step log in https://dataplatform.cloud.ibm.com/ with the same account of IBM Cloud.

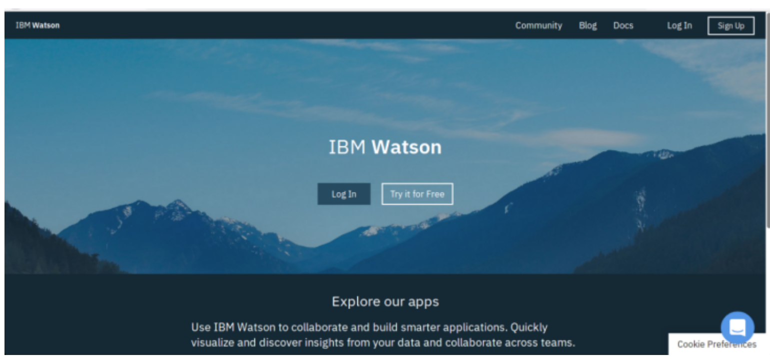


Figure C.1: Data platform website

As you can see, there are not Watson services activated yet. Click on New Service and in the next list click Add in Visual Recognition option.

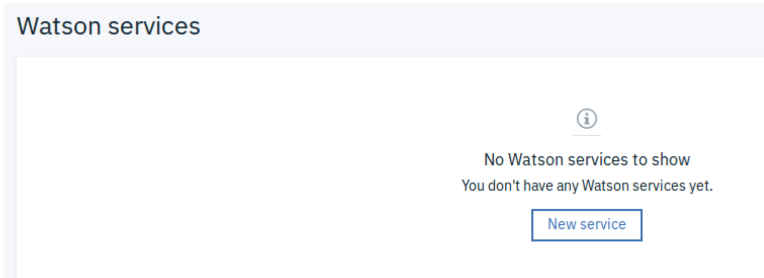


Figure C.2: Active Watson services

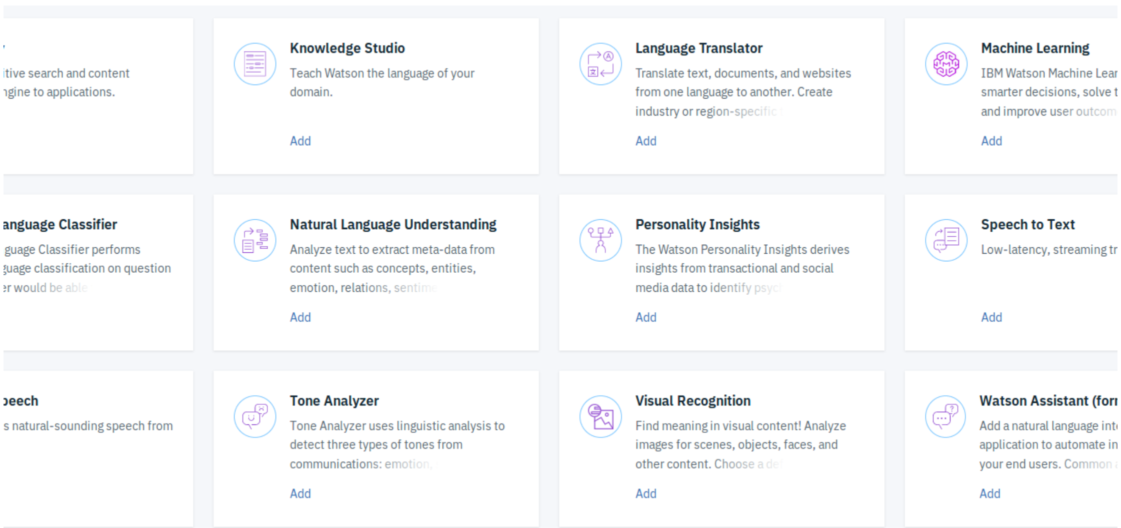


Figure C.3: Selection visual recognition option

Then select Lite Plan and create the service. After that a confirm message will appear, if all it is fine click in Confirm.

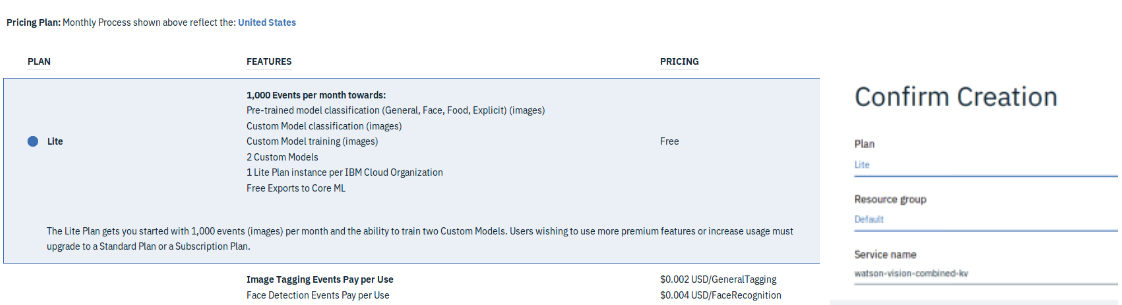


Figure C.4: Service creation

As you can see the service created watson-vision-combined-kv now is listed in the Visual Recognition services created. The next step will be click on Launch tool.

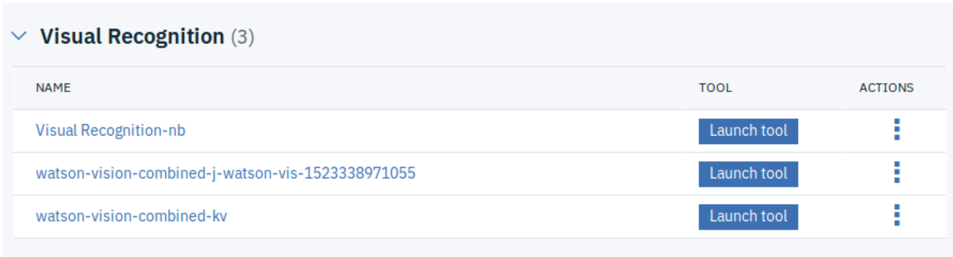


Figure C.5: Launching Watson service

This tool offers the next model options. Custom model is which we are interested in. Here is where you can create a custom model loading your own classes which will be identified in the test images. So click in Create Model and provide a name to the project.

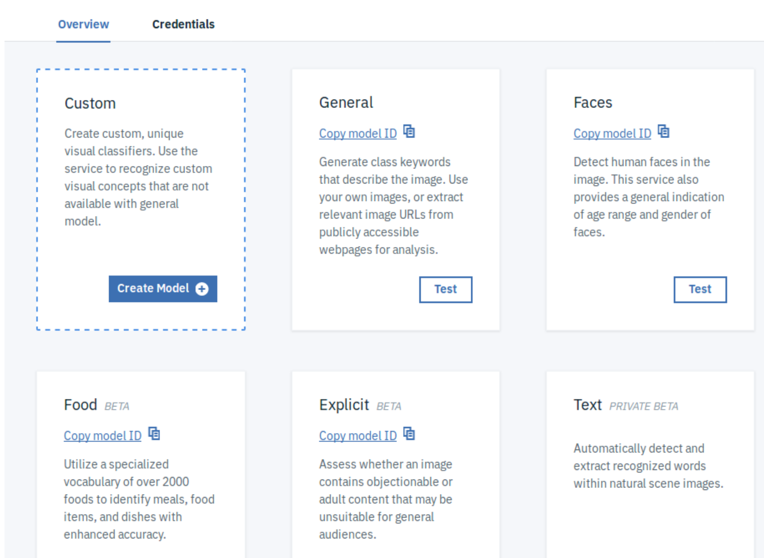


Figure C.6: Custom model creation

The warning message indicates there are not Watson service instance selected to the project. So click on here link in the message and then in Watson Studio option click *Try it for free* and select *Lite option.* After that, create and confirm the creation of the Watson service instance. Finally click on refresh link in the warning message. Define a name project and select the Watson service instance that have the same name of Watson Recognition Service created.

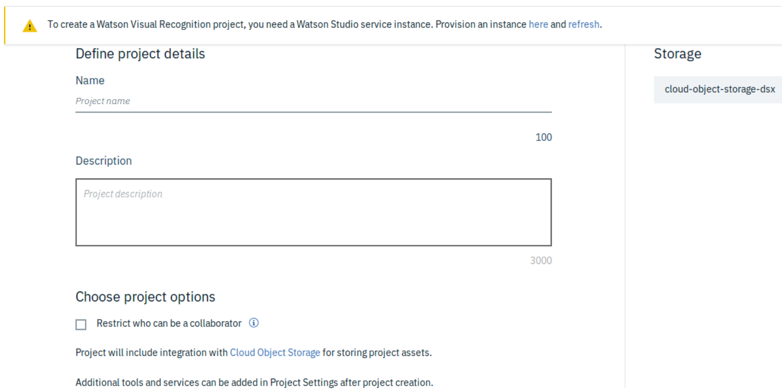


Figure C.7: Custom model settings

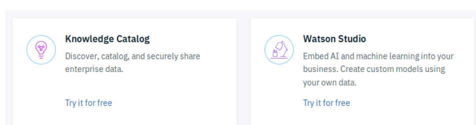


Figure C.8: Selecting Watson service instance

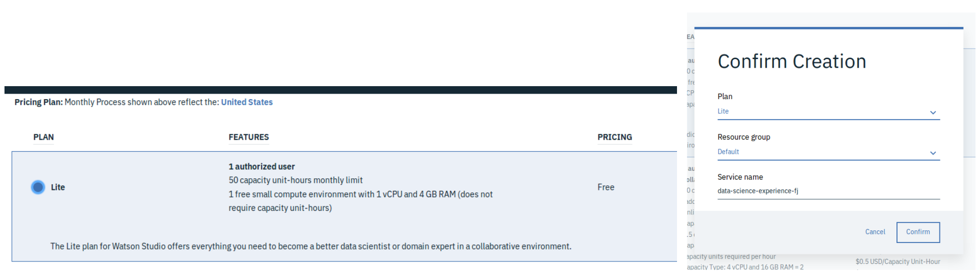


Figure C.9: Selecting lite plan

Once you finished all the steps, and click on Create the tool is ready to upload the zip files with maximum 250 images per zip. After all the needed zip files are loaded your model is ready to train it.

Figure C.10: Defining name project

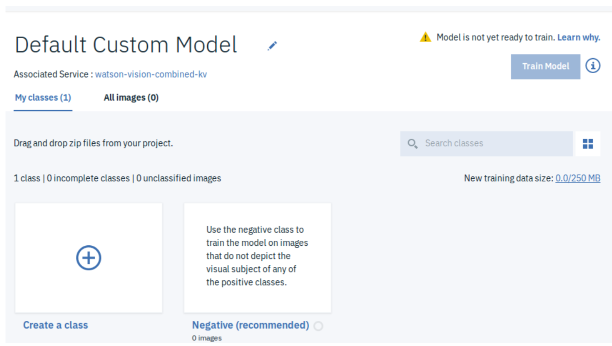


Figure C.11: Custom model tool

After that, click on Browse to select or drag the zip files containing the images (jpeg or png extension) to use to training the model.

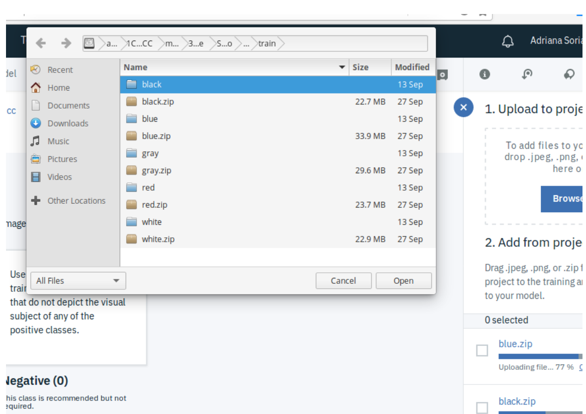


Figure C.12: Selecting zip files

Once the files are uploaded to your cloud object storage select the zip files and add them to the model. When all the added classes had a green circle, means all the images were added. To confirm, you can select the class to view the images in that class. Now your model is ready to train, as indicated on the label. When your ready train the model. In this case, after 57 minutes of training, this model was ready to test.

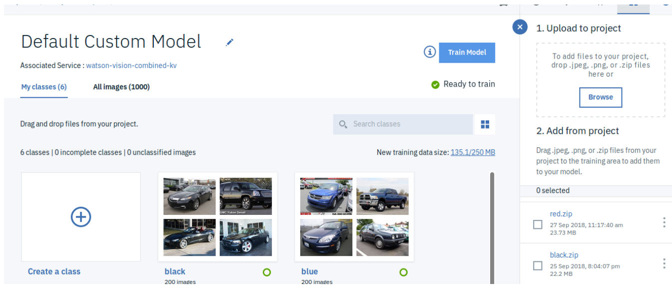


Figure C.13: Model ready to train

When the training is complete, use the link to view and test the model.

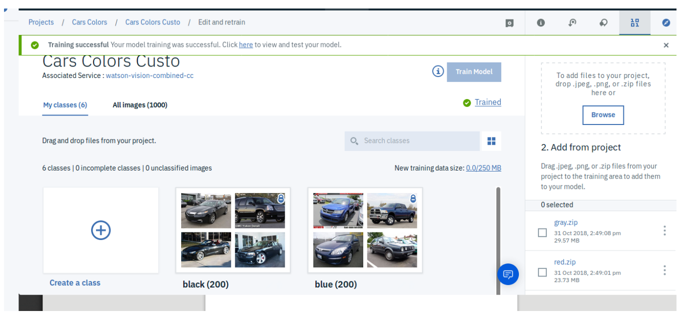
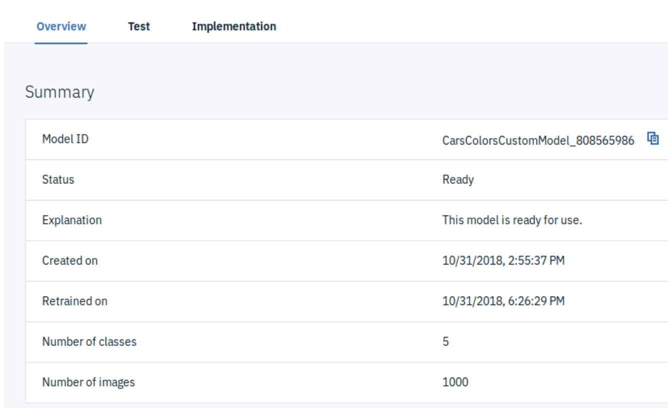


Figure C.14: Model trained

The overview tab includes a summary and a list of the classes.



Model ID	CarsColorsCustomModel_808565986
Status	Ready
Explanation	This model is ready for use.
Created on	10/31/2018, 2:55:37 PM
Retrained on	10/31/2018, 6:26:29 PM
Number of classes	5
Number of images	1000

Figure C.15: Model overview

On the test tab you can you can add images to analyze.

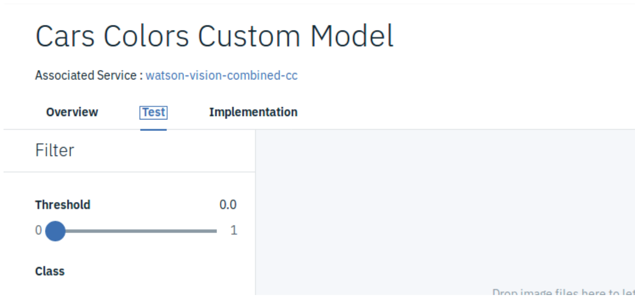


Figure C.16: Test model tool

And here are the results of showing the matching classes and the confidence score for that match. On the implementations tab, you will find the code snippets to use this model in your applications.

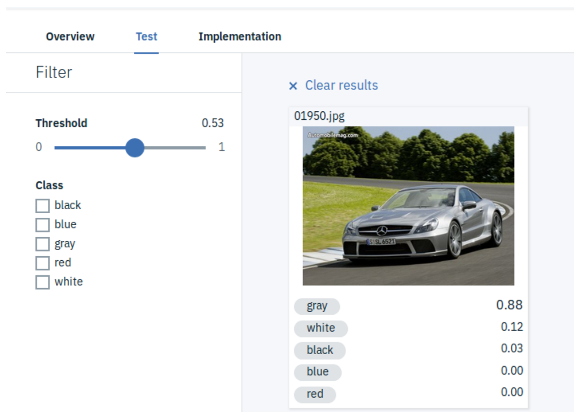


Figure C.17: Example of testing

The next image is an example of a python application using this model. In this application, you select and analyze one image per time. You can find source code here: https://github.com/Adrinems/VisualRecognitionIBMApp.

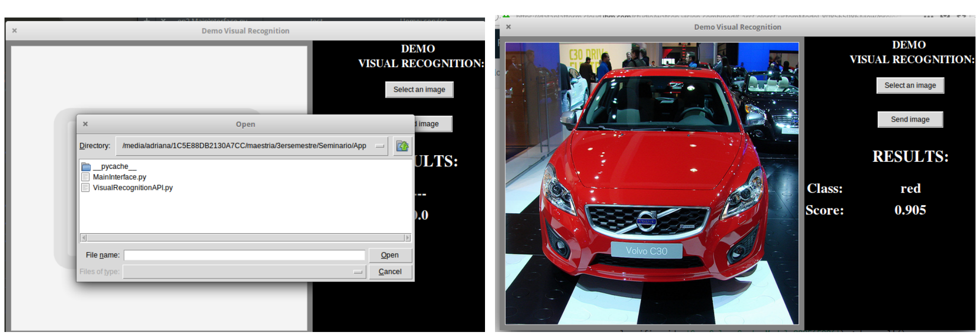


Figure C.18: Python application

Appendix D

IBM Cloud: Watson Machine Learning

Create a new project

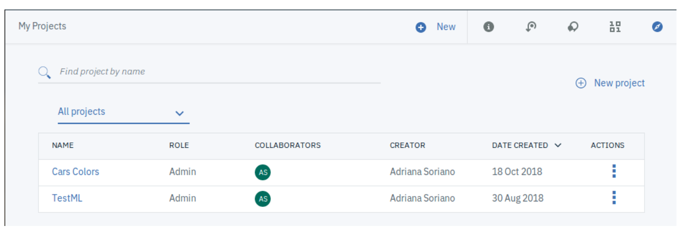


Figure D.1: New project Machine Learning Watson

Select Deep Learning type project

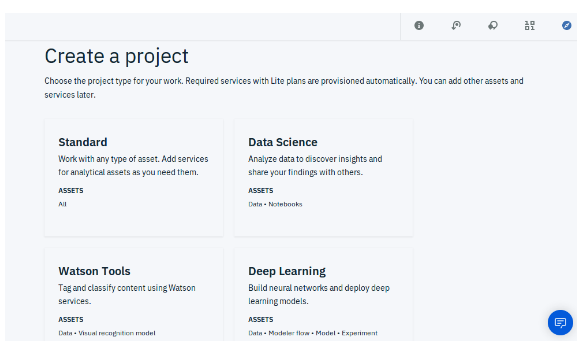


Figure D.2: Deep Learning project selection

Select the corresponding region, in my case is US South.

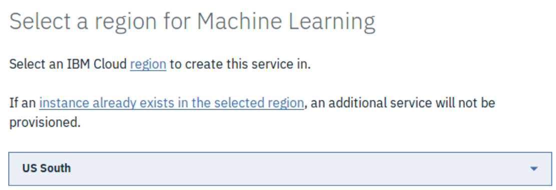


Figure D.3: Region configuration

Define the name and description (optional) of the project.

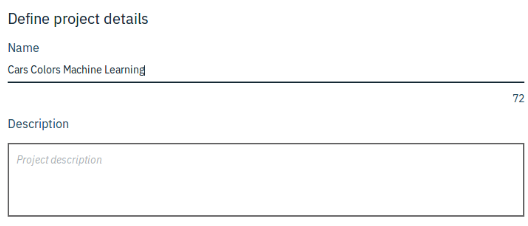


Figure D.4: Name configuration

Below, select the storage service. When a machine learning service is created, its name is gotten by default. If you have not created you will have to add a new service instance. Then, an image like Figure C.5 (right side) will appear to finish configuration step. After that, click on create button.

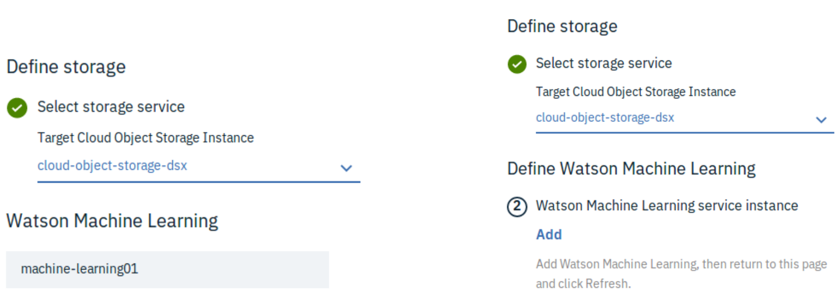


Figure D.5: Storage configuration

Now in Assets option create a new Model.

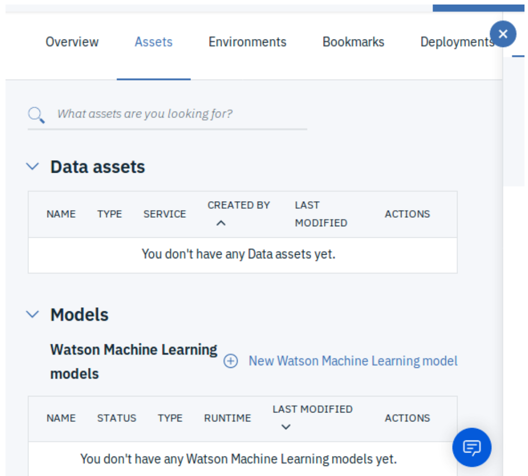


Figure D.6: Creating new model

Insert the name of the model and select the machine learning service to use.

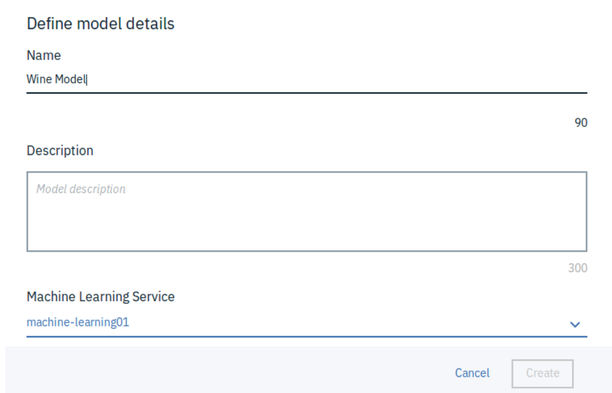


Figure D.7: Name model configuration

Select Model builder in model type. Select your runtime environment. In case you have not created yet, will be shown on screen to add one. Click the card labeled Manual (This will cause model builder to manually select an algorithm to implement machine learning technique.)

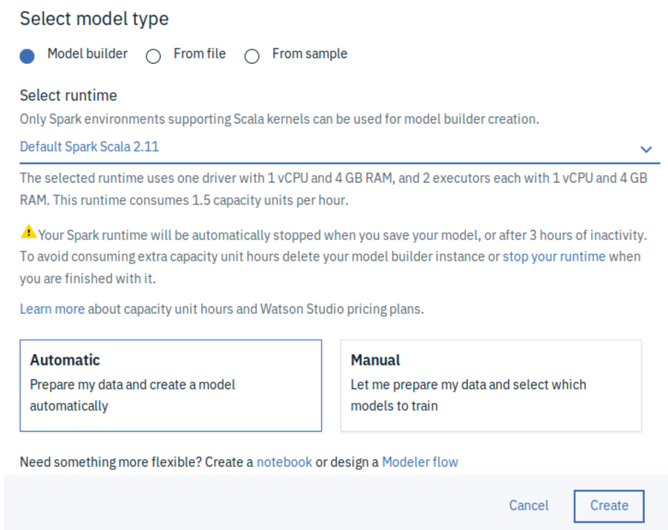


Figure D.8: Model type selection

Open a new window in data platform and click in Community option. For this example we will use Wine Recognition, so search wine and select this dataset.

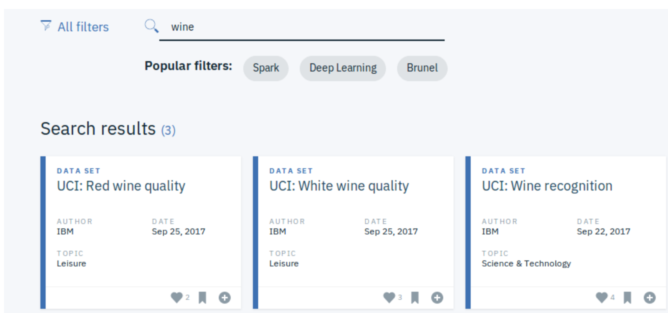


Figure D.9: Dataset searching

Click in add button in dataset and add to the project we are working in.

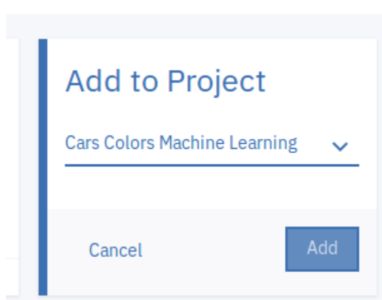


Figure D.10: Upload dataset to project

Return to the previous page where the model was created. If dataset was corrected added, it will appear in Select data asset step. Select it and click next.

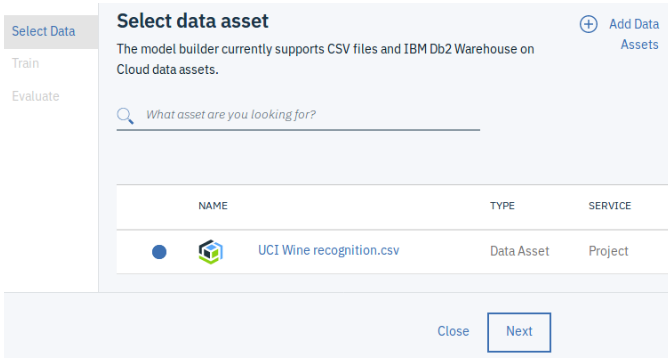


Figure D.11: Dataset selection in model

After this step, it will be explained roughly how the dataset is constructed. It is constructed in base of the results of a chemical analysis of wines grown in the same region in Italy, but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines The 13 attributes included in dataset are:

1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity

11. Hue

12. OD280/OD315 of diluted wines

13. Proline

As you can see, this is a classification problem.

In model builder, you can create three kinds of machine learning model:

- Binary classifier: Classifies data into two categories
- Multiclass classifier: Classifies data into multiple categories
- Regression: Predict a value from a continuous set of values

Builder suggest to use Multiclass classifier model because it is a classification problem and we have 3 categories. For each kind of model, you can choose from multiple algorithms or estimators to implement the technique. It will appear in the right side.

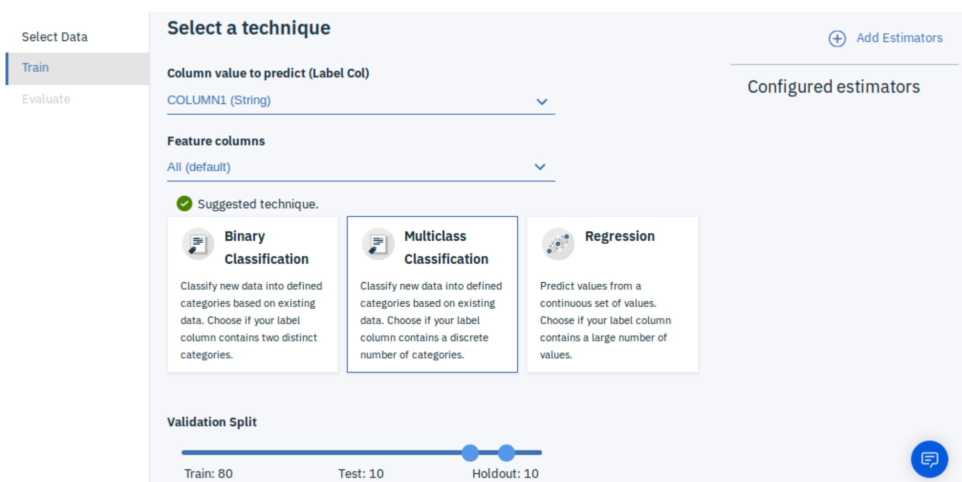


Figure D.12: Configuring model builder

Now select all the estimators you can use to train your model. In bottom page, configure the validation split (image D.12). Select the percentage of train, test and holdout for the model.

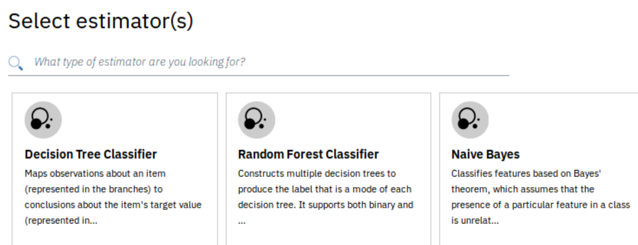


Figure D.13: Estimators

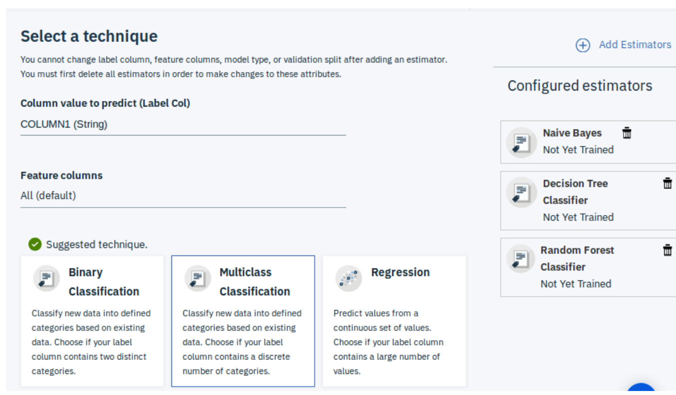


Figure D.14: Selecting estimators

Next step is to train your model. In image D.15 the training is being executed. You have to wait until the model finish. In this case took 15 seconds to train Decision tree, 35 seconds for Naives Bayes and 24 seconds for Random forest.

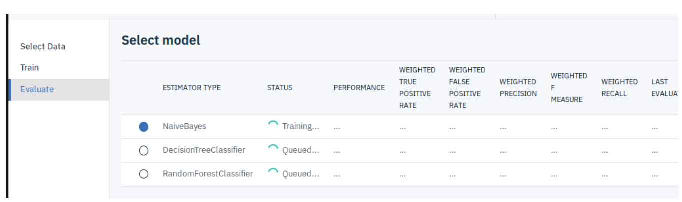
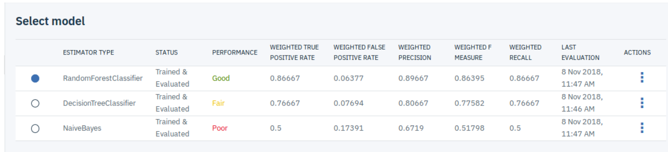


Figure D.15: Training model

Once training has finish, select the estimators which you wants to continue for testing. For this example we choose Random forest because it was with best results.



Select model

	ESTIMATOR TYPE	STATUS	PERFORMANCE	WEIGHTED TRUE POSITIVE RATE	WEIGHTED FALSE POSITIVE RATE	WEIGHTED PRECISION	WEIGHTED F MEASURE	WEIGHTED RECALL	LAST EVALUATION	ACTIONS
	RandomForestClassifier	Trained & Evaluated	Good	0.86667	0.06377	0.89667	0.86395	0.86667	8 Nov 2018, 11:47 AM	
	DecisionTreeClassifier	Trained & Evaluated	Fair	0.76667	0.07694	0.80667	0.77582	0.76667	8 Nov 2018, 11:46 AM	
	NaiveBayes	Trained & Evaluated	Poor	0.5	0.17391	0.6719	0.51798	0.5	8 Nov 2018, 11:47 AM	

Figure D.16: Finished model training

The next image shows an resume of the training step with the selected estimator.

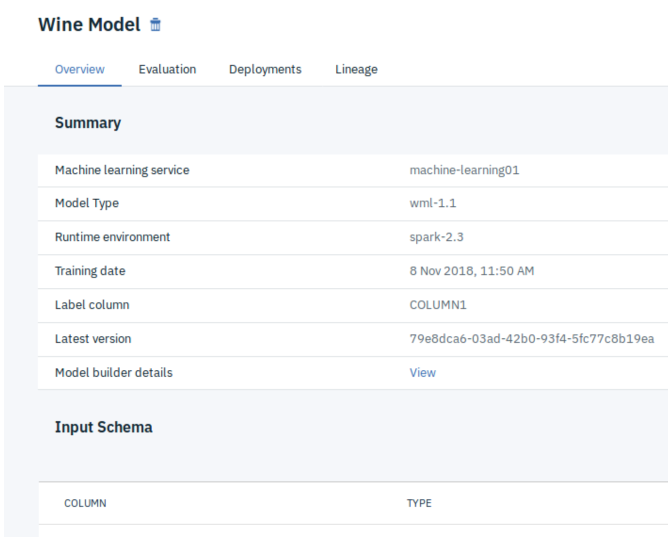


Figure D.17: Resume training model

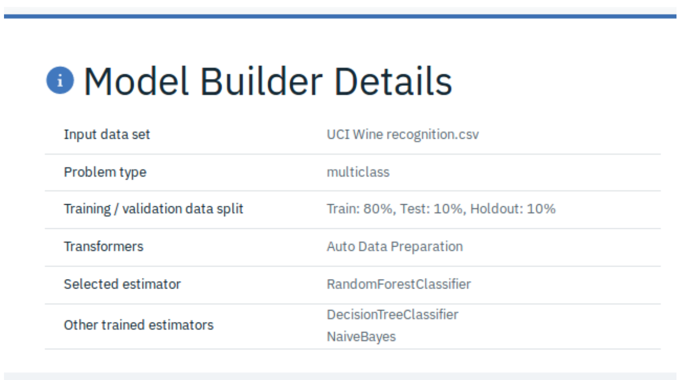


Figure D.18: Model builder details

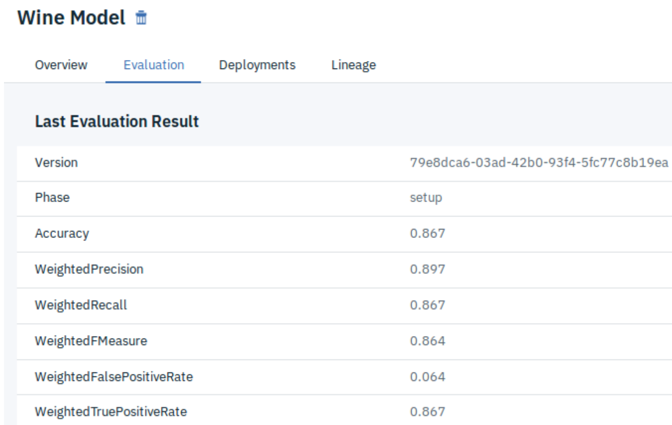


Figure D.19: Model Evaluation

Select Deployments tab and add a new deployment.

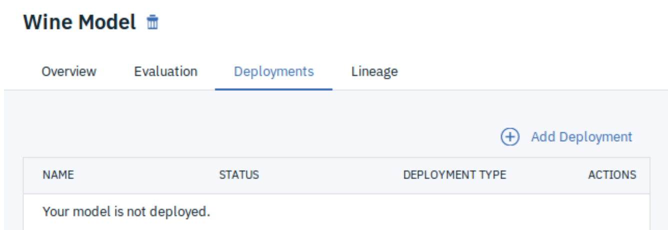


Figure D.20: Create model deployment

Define a name, a description and deployment type and save it.

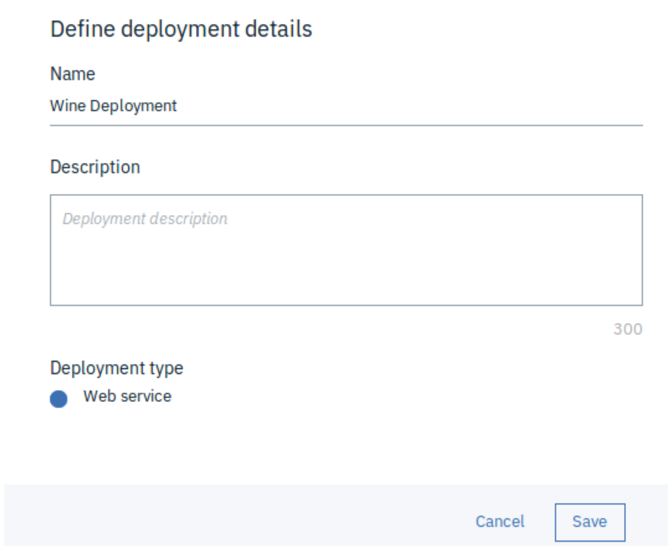


Figure D.21: Define deployment details

When deployment has been correctly added you can select it or click on Actions menu and click on view and start your testing.

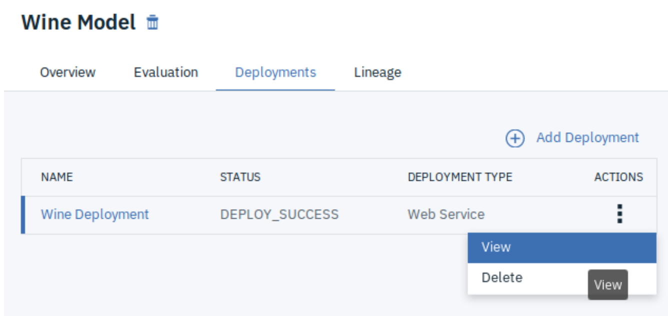


Figure D.22: Start deployment

For testing is necessary to introduce the values for the 13 categories which will be evaluated. After that you click on test and wait for results.

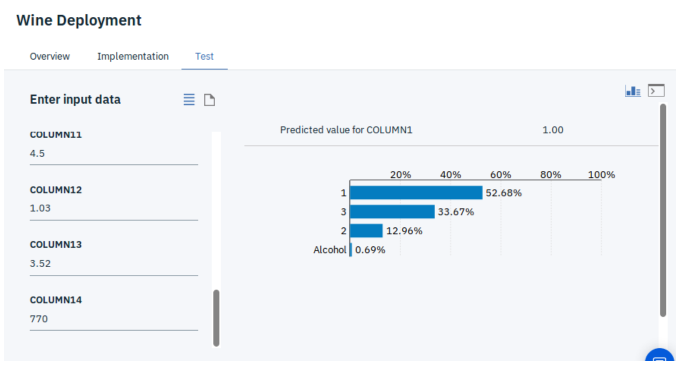


Figure D.23: Results testing machine learning with random forest algorithm

Bibliography

[1] CS231n. *Convolutional neural networks for visual recognition.* 2016. http://cs231n.github.io/convolutional-networks/.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[3] C. Gulcehre-http://deeplearning.net/.

[4] A. González http://cleverdata.io/que-es-machine-learning-big data/.

[5] ImageNet http://image net.org/.

[6] T. M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[7] Waseem Rawat and Zenghui Wang. *Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review.* Massachusetts Institute of Technology, 2017. http://cs231n.github.io/convolutional-networks/.

[8] Berkeley Artificial Intelligence Research.

[9] E. Roberts. *Neural Networks.* 2016. https://cs.stanford.edu/people/eroberts/courses/soco/projects/neuralnetworks/History/history1.html.