

# Deep Learning Method based on Big Data for Defects Detection in Manufacturing Systems Industry 4.0

Ashraf Abou Tabl<sup>1,2,\*</sup>, Abedalrhman Alkhateeb<sup>3</sup>, Waguih ElMaraghy<sup>1</sup>

*1 Faculty of Engineering, Department of Mechanical, Automotive & Materials Engineering (MAME),  
University of Windsor, Windsor, ON, Canada.*

*2 School of Business & IT, Data Analytics Department, St. Clair College, Windsor, ON, Canada.*

*3 School of Computer Science, University of Windsor, Windsor, ON, Canada.*

\* Corresponding author: aboutaba@uwindsor.ca

---

## Abstract

Due to the technological advancement in Today's manufacturing systems, a large amount of data is generated in different volume, velocity, and variety of kinds. Extracting information from these data and make a real-time decision is a big challenge to the current manufacturing systems. This study presents a novel model that converts the iFactory learning facility into a fully Industry 4.0 (I4.0) manufacturing system. To achieve this purpose, we utilized the cyber physical system (CPS) components and sensors, the Internet of Things (IoT), deep learning methods, and cloud computing to fully meet the I4.0 enablers. Cloud computing is utilised in two phases: (1) during the model training phase to hold a large amount of product image data collected from the inspection station, and (2) during the execution of the model.

The core learning model is based on a convolutional neural network (CNN) that is trained from the captured product images in the production line to predict the defective items in the line. The model was initialized by Resnet method and optimized to improve the learning rate and reduced loss function. The supervised learning model achieved high accuracy prediction performance up to 96.75% in the real-time decision making process. The model was able to extract the feature map of the normal non-detective product and use it to improving the accuracy and reducing the traffic between iFactory station and the cloud server. The model exploits the parallel computing big-data framework to achieve a real time decision making. The model can be applied to the current system and adopted as with all it is functionalities for the newer systems.

**Keywords:** *Big Data; CPS; IoT; Cloud Computing; Smart Manufacturing; Industry 4.0; Machine Learning; Deep Learning; Convolutional Neural Network.*

---

## 1- Introduction

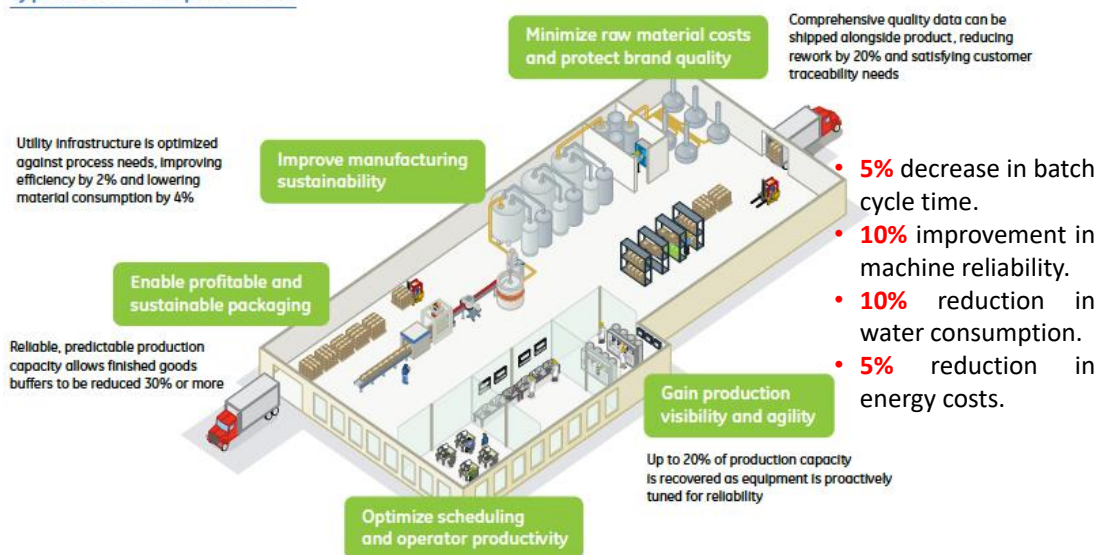
In today's competitive manufacturing environment, manufacturers are facing challenges related to dealing with big data, such as how to rapidly make decisions to improve productivity, quality, and system sustainability[1],[2], [3]. Manufacturing facilities are not ready to manage and obtain the advantages of big data due to a lack of decision making built-in metho [4]. These advantages are seen in Fig. 1. Manufacturers realizes the importance to adopt smart decision making methods for big data, and also respond and adapt to the continuous technological developments, dynamic markets, and demand fluctuations [5]. Recent

technological manufacturing developments are encouraging the use of CPS, whereby information is gathered from all related sources. It started with a project that is funded by the German government for advanced manufacturing ideas in 20 [3]. The paradigm of the fourth industrial revolution is based on CPSs[1], [2] , so it will entail new sorts of complexity, namely, the complexity of structure, data, products, and interaction. The new revolution involves using new technologies, including CPSs, cloud computing, big data, and the IoT, which is sometimes referred to as the internet of services (IoS).

Big data analytics is a relatively new research area and a key enabler of Industry 4.0. The IoT and cloud computing are highly recommended to be utilised in the manufacturing decision making processes. Big data research is currently concentrated on capturing, pre-processing, processing, and analysing big data [6] rather than other aspects that have relatively less value, such the traditional storage techniques [7].

Current manufacturers are not fully utilising the scale of big data or engaging in real-time decision making, which limits success [8], and there are still significant challenges associated with the implementation of big data [9]. In [10] suggested that researchers should focus on big data algorithms development for manufacturing operations.

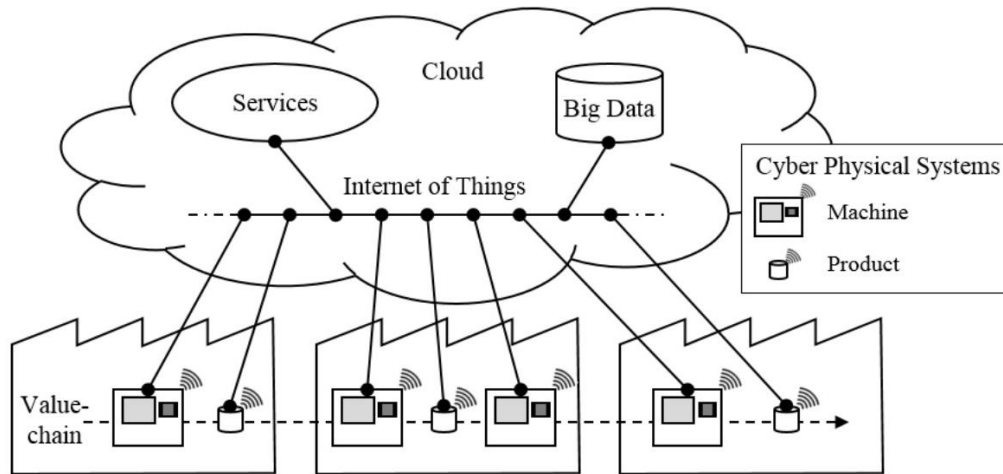
#### Typical Process Improvements



**Fig. 1. Typical Process Improvements [8].**

The sharing of components and services is vital to cloud computing and significantly contributes to Industry 4.0 [11]. The concept of cloud computing is that everything, even processing resources, such as CPUs and graphics processing units (GPUs), can be shared as services.

In [12] applied a CPS model to an automated electric vehicle by applying a machine-learning approach. The method involved optimising various CPS parameters to improve various performance objectives, such as drivability and energy, dynamic vehicle performance, along with differing driving styles. The results validate the effectiveness and feasibility of the proposed CPS-based approach for three driving styles: conservative, moderate, and aggressive [12] . Fig. 2 shows the structure of a CPS system with all enablers.



**Fig. 2 CPS system with all enablers [13].**

The iFactory was utilised in this research. It is a state-of-the-art learning factory located in an intelligent manufacturing system laboratory that is one-of-a-kind in North America [14], as shown in Fig. 3. It simulates the innovative physical integration of varying production lifecycle phases: iDesign, iPlan and iFactory. iFactory is a modular manufacturing system that can be reconfigured to have a different line of production structure. The main goal of this study to transfer the iFactory to be fully adopting the Industry 4.0 revolutionary enablers by using IoT devices such as cameras and sensors, these devices must be able to stream reading data into the cloud for real-time decision making based on machine learning. Based on the proposed method, with the help of Artificial Intelligence (AI), we improved iFactory skills to monitor any unusual events, continuously learning from the available and newly generated data, and take a decision such as predict the defective product.



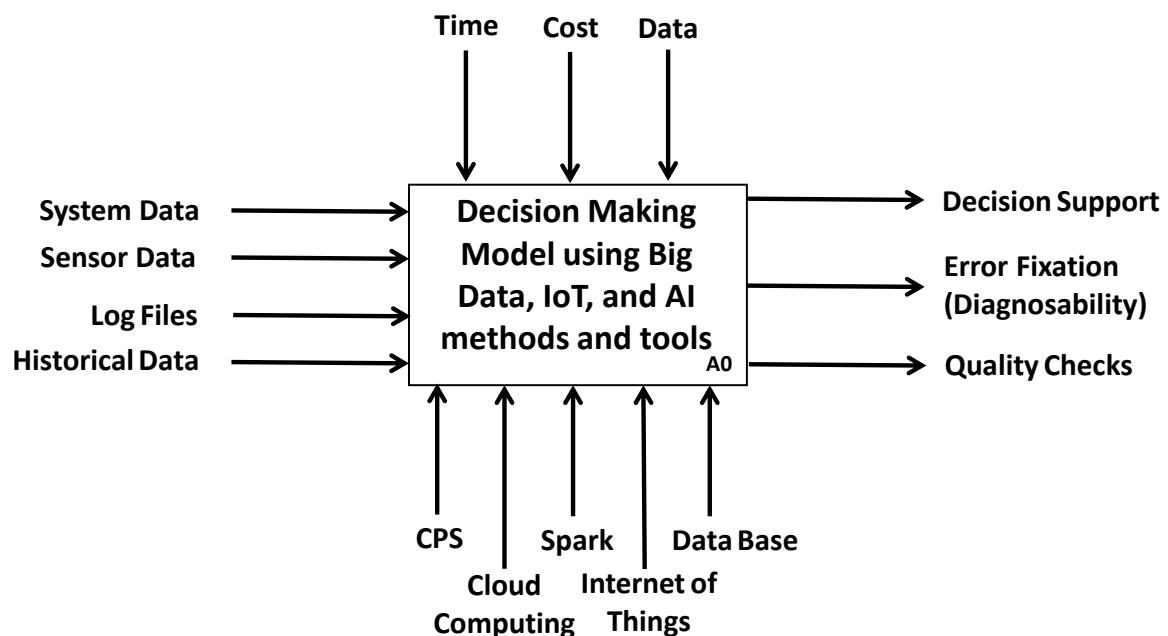
**Fig. 3 iFactory.**

## 2- Materials and Methods

Using the current inspection method, the iFactory inspection station camera targets whether the positioning of each product is correct or incorrect. This is done currently in the iFactory using the PLC programming unit. The objective of this study is to utilise capabilities of I4.0 by implementing all its enablers. This process is performed using a big data analytics platform with the guidance of the I4.0 concepts. The process involves capturing images of the product and transferring them to a big data platform on a supercomputer. Image pre-processing and decision making is done remotely on the supercomputer. Then the results are instantly transferred to the product station. Utilizing a big data platform has become a necessity due to the increased volume of data and the urgency of taking real-time decisions.

The product quality is manually classified into two main classes: a “normal” class in which the image of the product does not have any evidence of a defect, and a “bad” class in which the image explains some defects in the product. A CNN based on TensorFlow was utilised as an efficient classification tool for making such decisions. Moreover, due to the fine details and similarities in the dataset, transfer learning was used based on the Resnet-50 pre-trained model by uploading its weight matrix to our model [15].

Fig. 4 shows the high-level IDEF0 graphical function model for smart manufacturing systems. It depicts the main activities of the proposed method including all the inputs, outputs, mechanisms, and controls. The inputs include multiple big data sources, such as system data, log files, and sensor data. While the outputs include quality assurance, the error corrections, and the decision support actions. The functionalities include big data tools, utilizing supercomputer resources, CPS capabilities, and IoT techniques.



**Fig. 4 High-level IDEF0 graphical function model for smart manufacturing systems.**

The details about the IDEF0 used for this part of the research are shown below in Fig. 5. It starts with data mining to analyse the input data, such as sensor data, historical data and log files. The system uses product status (i.e., acceptable or defective) to detect and label the

classes (i.e., defective labelled as 0 and non-defective products labelled as 1) using the captured product images. The inspection station camera captured a few thousands of images which is still not enough for proper training. Therefore, Augmentation pre-processing on these images was essential to increase the number of images from 5,000 to 20,000 (10,000 images of each class). The Augmentation consists of rotating, scaling, and resizing the captured images. The next step is transfer learning using CNN that is built on resnet-50 and a 3x3 kernel matrix. The activation functions of CNN are ReLU and Softmax, and the result of this step is to extract the discriminative feature map. We used Sickitlearn library based on TensorFlow to build the fully-connected layers network. Running the classification for predicting the classes.

The classification or the prediction model, as shown in Fig. 6 below, consists of the training step by splitting the samples into 70% for training, 20% for validation, and 10% for testing the model. The training phase changes the weight of the fully connected CNN throughout the forward/backward propagation process to improve the learning rate. Then the model is tuned by grid-search the batch size and the number of epochs. The model is optimized using the ADAM method [16] followed by model regularisation using different rates of dropout. Validating the model, then testing is the final stage to ensure the proper performance with no under-fitting or over-fitting in previous stages. The testing phase is a classification process on the never seen samples in the learning phase.

#### **A. Scope and Applicability to Upgrading Existing Manufacturing Systems**

The scope of this part of the paper includes manufacturing systems with low, medium and high production volumes for product families and variants. It also covers big data concerning variety (i.e., structured, non-structured or semi-structured), volume (e.g., up to terabytes and zettabytes), and velocity (i.e., real-time data). This model can be used to upgrade the current manufacturing systems which have I4.0 limitations, and also should be used on the newer released manufacturing systems by equipping them with the required communication hardware and software CPS system, which is vital to make the IoT system parts communicate with the central supercomputer. The newer systems need extra hardware and software features such built-in IoT cameras and sensors for better communication and helped the decision making process. These cameras and sensors can have specific filters based on the analysed feature mapped from the learning system to reduce the traffic and enhance real-time decision making. The system can also record some statistics for maintenance and overall performance predictability, EigenInovations is a good example of this process, as shown in Fig. 7 [17], but still, there is a lack of published methods in the literature.

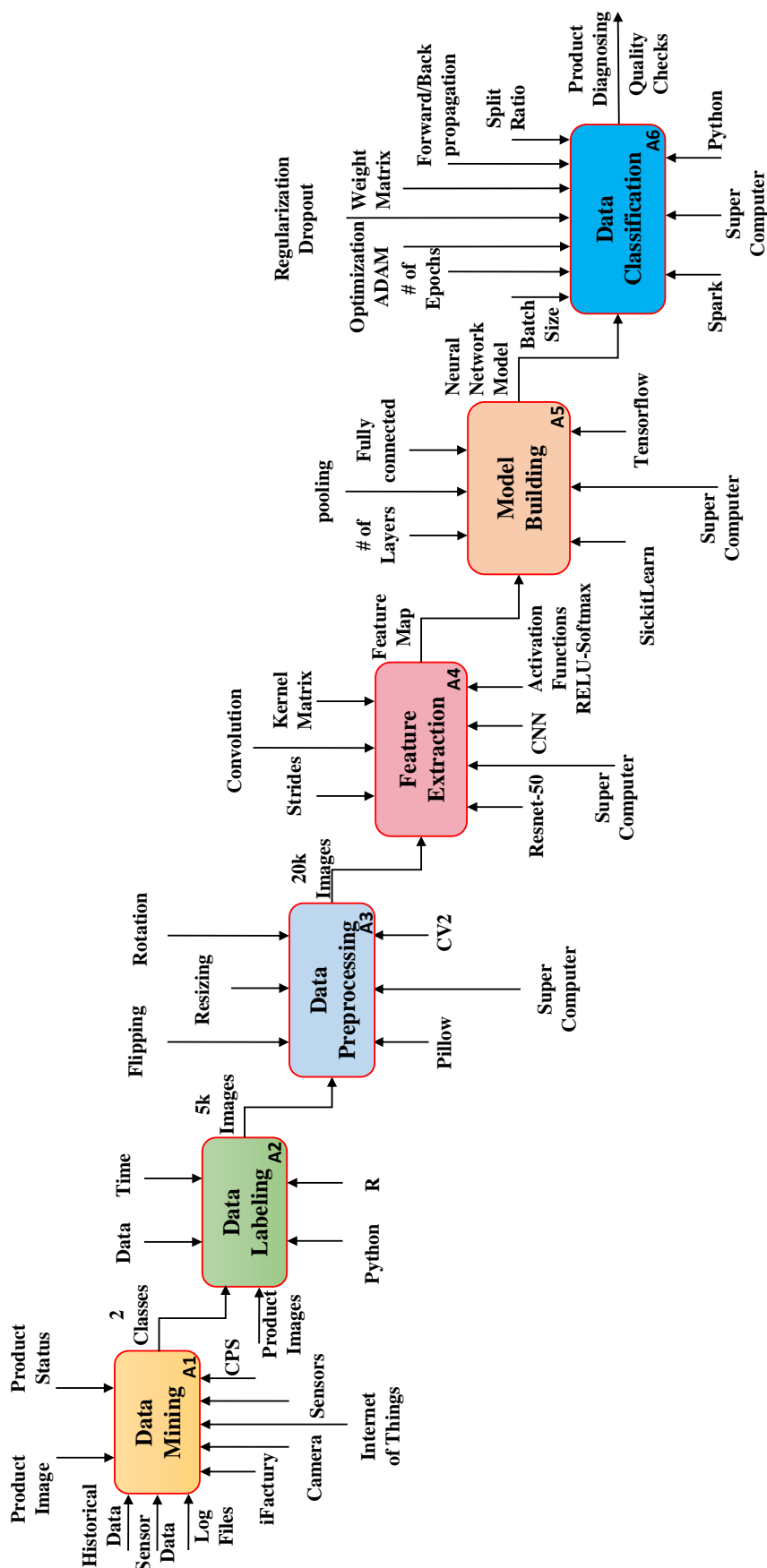


Fig. 5 Details about the IDEF0 used in this study.

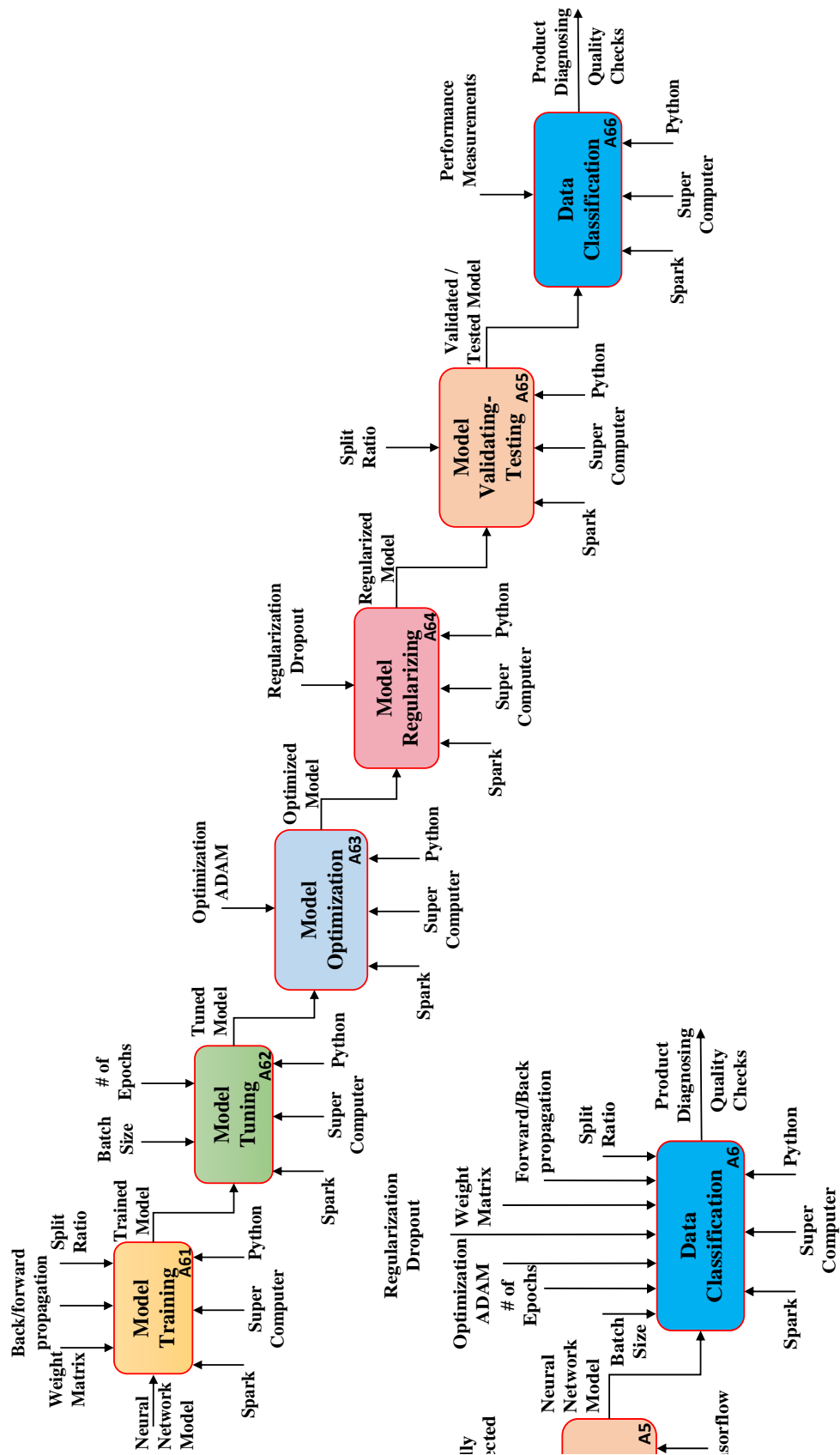
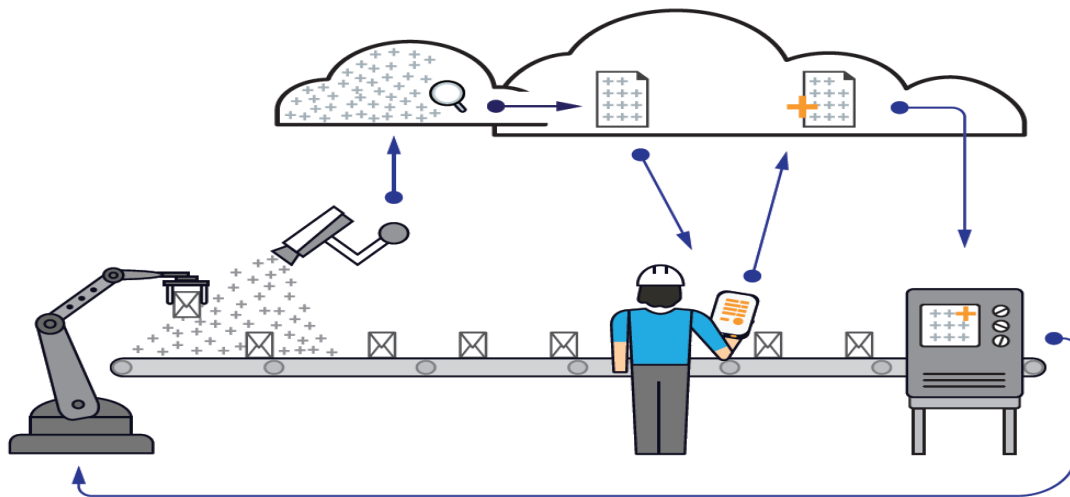


Fig. 6 Data classification steps.

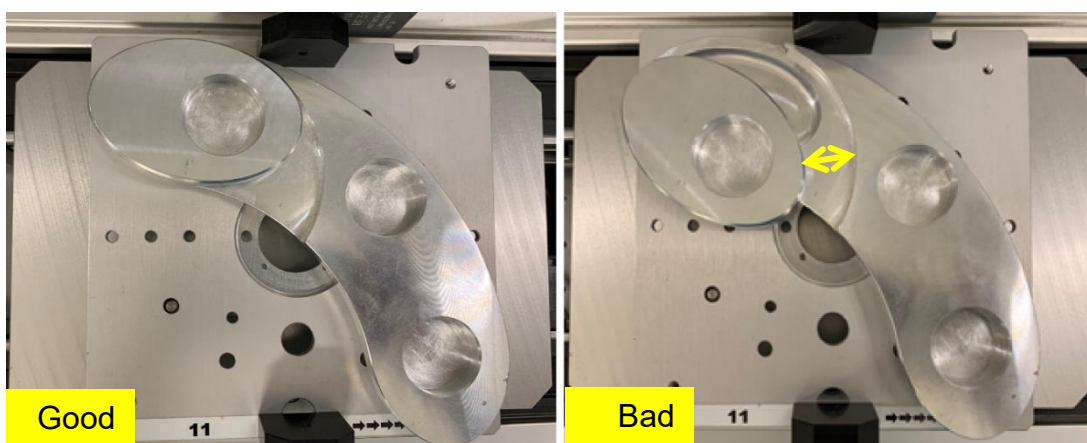


**Fig. 7 EigenInnovations' new manufacturing system [17].**

### **B. Data Acquisition**

In this phase, the camera (1,024 x 1,024 DPI resolution) is mounted to a base in the inspection station of the iFactory and used to capture product images. These images are in red, green, and blue (RGB) format. A total of 20,000 images were acquired to build a sufficient dataset. A supercomputer is used to split the acquired images into training, validation, and testing datasets. The training images are labelled using natural language processing with an accuracy level of 99%. After the datasets are pre-processed then they are fed as the input of the machine-learning model.

Fig. 8 illustrates the two classes. The left image is for the normal class, while the right is for the defective one. Since the used samples are labelled in the training, this model is considered as a supervised machine-learning problem. The normal class is labelled as “good” (coded as 1 in the system), and the defective class is labelled as “bad” (i.e., coded as 0). Each class consists of 10,000 images which make the model be a balanced classification model.



**Fig. 8 Samples of the normal (good; 1) and defective (bad; 0) product images.**

### **C. Pre-processing**

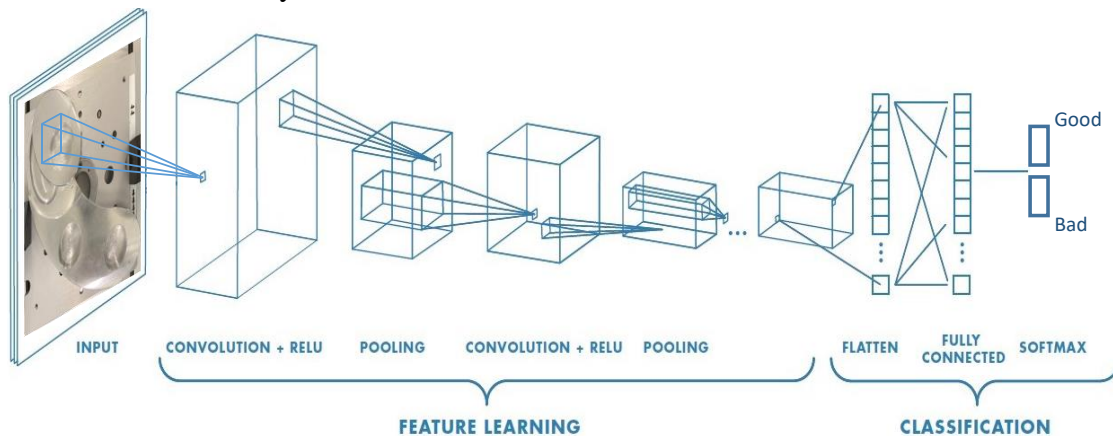
The pre-processing phase prepares the images to fit within the machine-learning model to extract the best features quality. In this phase, the images are normalised by scaling them



down to 512x512 pixels. Then all image channel values are normalised in the range between 0 and 1. This is achieved by dividing all pixel values by the highest pixel value, which is 255. The purpose of this normalisation is to remove any variation in channel values and accelerate the convergence of the machine-learning model. In the second phase, image augmentation is applied to increase the number of images and minimise the chances of overfitting. Augmentation is achieved by randomly and horizontally flipping a certain proportion of images and rotating them by 45 degrees. The features acquired for each batch are captured by cropping the image from the centre and rescaling it to 224 x 224 pixels. The image labels are either “normal” or “bad” and represented in a one-hot encoding vector by 1 or 0, respectively.

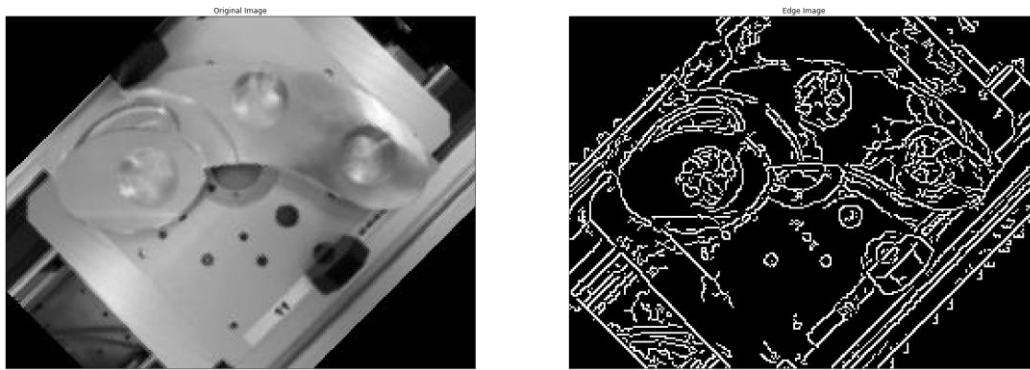
#### **D. Model Architecture**

The model has three phases: first is reading the dataset that is stored on a supercomputer. The second phase is pre-training the model weights using the initial values that are generated using Resnet-50. The Resnet-50 expands the machine-learning model to be deeper. Our model is trained from end-to-end layers (without freezing layers) to capture the fine details of each class that are called the discriminative features from the feature space. Resnet-50 consists of a convolution layer, a max-pooling layer, a ReLU activation function, as given in Equation 2, and uses dropout some layer as a regularization technique. The first layer is the input layer, and the final layer is a fully-connected class layer. The SoftMax nonlinearity activation function given in Equation 3 is performed with a binary cross-entropy loss function. The state-of-the-art method called ADAM (a method for stochastic optimisation method), a stochastic gradient descent method is used to optimise the loss function. Fig. 9 depicts the structure of the CNN layers.



**Fig. 9 Sample of CNN layers [18].**

The reason for using a CNN is its feature extraction efficiency that is achieved by multiplying images with a particular kernel (image filter). The first layer of CNN is the convolution layer, which extracts the discriminative features using the dot product of the image with a filter matrix known as a kernel that has a specific size. The spatial information is preserved in the outcome of the dot product in the pooling matrix that forms the feature map. The kernel filter slides from the left to the right of the image with one or two overlapping pixels or stride. We select the kernel size to be 3x3. Using different kernel values (filter) on the same image will generate different feature maps. Fig. 10 shows the extracted feature map when a “canny” filter is used.



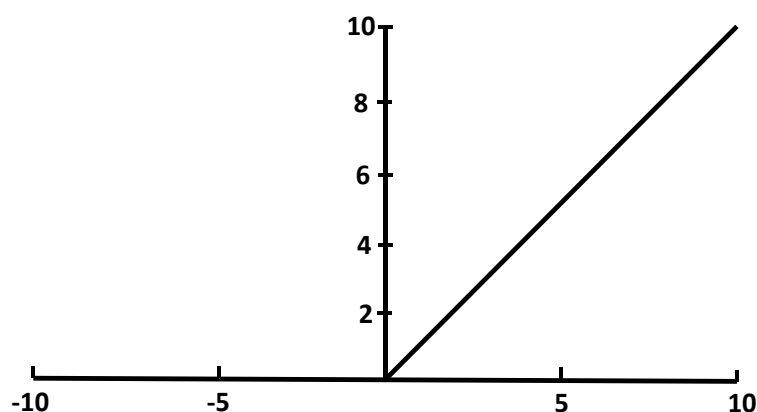
**Fig. 10 Product image and canny filter output.**

The second layer is the pooling layer, and the primary purpose of this layer is to train the model to identify the object, regardless of its position in the images. For example, if a max-pooling layer is used with 2x2 for every four cells, then the maximum numerical value is computed and inserted into the pooled feature map. Then the activation functions, such as ReLU and SoftMax, minimise the produced number by the former layers. Each number which is generated by the activation functions at each layer are stored in the weight matrix. The purpose is to reach the weight matrix that best minimises the loss function, which a binary cross-entropy loss function is given in Equation 1:

$$CE = - \sum_{i=1}^{c'=2} t_i \log(s_i) = -t_1 \log(s_1) + (1 - t_1) \log(1 - s_1) \quad (1)$$

The loss function measures the variation between the value of the sample and the predicted value. The activation function we utilised is Rectified Linear Unit ReLU. As shown in Fig. 11, ReLU is an element-wise operation (i.e., it is applied per pixel) and set all negative pixel values in the feature map to be zero. The purpose of using ReLU is to introduce nonlinearity to the CNN model due to the nature of the data in real life is being generated by non-linear functions and the fact that convolution is a linear function [19].

$$\text{output} = \max(\text{zero}, \text{input}) \quad (2)$$



**Fig. 11 RELU (Rectified Linear Unit).**

The weight matrix keeps updating itself using forward and backward propagation. The last fully-connected layer with two classes is used with SoftMax [20], as shown in the following equation

$$\alpha(x_j) = \frac{e^{(x_j)}}{\sum_i e^{(x_i)}} \alpha(x_j) = \frac{e^{(x_j)}}{\sum_i e^{(x_i)}} \quad (3)$$

Where  $\alpha$  is the output of the activation function,  $x_i$  is the weight matrix of layer  $j$  and  $x_j$  is the output of layer  $i$ .

The bad and good product classes are coded as 0 and 1 respectively. We used 227 layers for the sake of implementing transfer learning by the Resnet-50 pre-trained model. However, Resnet -50 was trained from scratch, and all weight matrix values were discarded for the sake of improved learning [15].

### E. Tuning the Model’s Hyperparameters

In this approach, the hyperparameters are tuned by observing the accuracy of performance. We used the well-known “Stop Early” feature when the accuracy score achieves 99%. The training is repeating 10 times (epochs) until the targeted accuracy have been completed. Different batch sizes of 20, 40 and 60 images per batch were used. The dropout rate value was set to be 0.2 to prevent overfitting and improve accuracy. The learning rate was tuned using a learning rate scheduler (a sort of grid-search method) with a base learning rate of  $1e-7$  worm up to 5 epochs (0.001) then dropped down to a minimum learning rate of  $1.E-15$ . However, the learning rate for which the highest accuracy level was achieved was 0.001. The optimiser used to minimise the loss function was ADAM.

### 3- Results

A supercomputer was utilised to test this application using Apache Spark, an open-source general-purpose cluster computing framework. Spark also provides an interface for programming clusters with data parallelism and fault tolerance. The model was running on a Compute Canada supercomputer known as Cedar in a cluster with four nodes. Each node has 16 cores and 100 GB of memory per CPU. The model was tested with batch sizes 20, 40 and 60. The results are shown in Table 1, which shows the model’s performance measurements, including accuracy and loss of classification.

**Table 1 Results (Accuracy and loss) with different batch sizes, Epochs, and Drop out.**

Batch size	Dropout # of Epochs	0.1			0.2		
		5	10	15	5	10	15
20	Accuracy	55.87%	76.85%	95.84%	80.68%	96.78%	96.75%
	loss	69.35%	51.82%	0.99%	43.18%	2.17%	5.61%
40	Accuracy	51.27%	72.56%	93.79%	73.23%	92.45%	94.99%
	loss	69.25%	51.61%	9.09%	40.41%	10.99%	0.25%
60	Accuracy	48.16%	60.96%	88.91%	55.78%	62.45%	89.41%
	loss	68.95%	65.41%	14.96%	69.32%	60.87%	21.63%

Accuracy was determined as the percentage correctly-classified images divided by the total number of images while the loss was defined as the least square function that takes the square root of all the summed-up squared the values differences between the prediction value and the actual value for all classified samples. By changing the number of nodes used to measure the scalability of the system, additional data can be processed in less time than ever.

### A. Performance

The performance was measured based on how many images can be processed per epoch per time. It was noticed that with increases in batch size, accuracy and training time decreased. However, using a small batch size takes a long time but results in high accuracy. When we used a batch size of 60 with 10 epochs, each epoch lasted approximately 7 minutes and 62.45% accuracy was achieved. However, when we used a batch size of 20 with the same number of epochs, each epoch lasted 20 minutes and 96.78% accuracy was achieved.

### B. Accuracy with different batch size and dropout

In this experiment, accuracy was measured based on the batch size with a dropout of 0.1. With the increase in batch size, the training time decreased and accuracy increased. The number of epochs was set to 5, 10 and 15. The best results (i.e., 95.84% accuracy) were achieved for 12 epochs with a batch size of 20. The best learning rate was 0.001. Fig. 12 depicts the results based on accuracy, batch size and number of epochs when the dropout rate is 0.1.

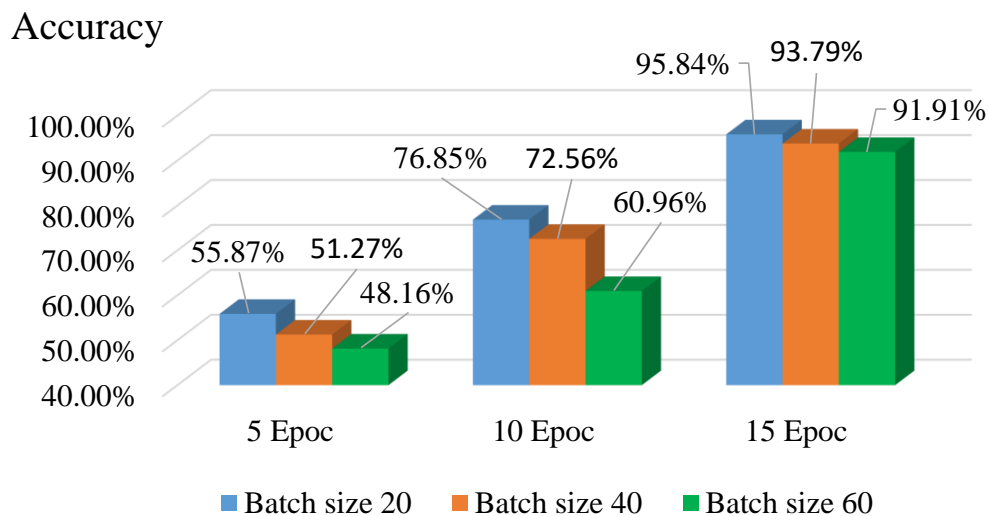
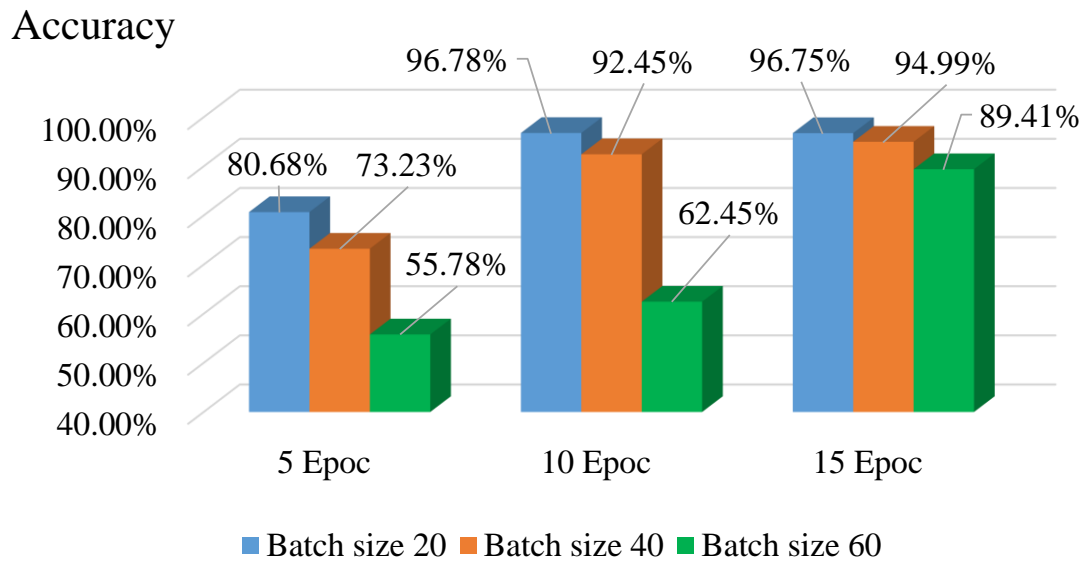


Fig. 12 The accuracy of the model using a different number of epochs and different batch sizes where dropout is 0.1.

In this experiment, accuracy was measured based on batch size with a dropout of 0.2. With increases in batch size, the training time decreased and accuracy increased. The number of epochs is set to 5, 10 and 15. The best results (i.e., 96.78% accuracy) were achieved when the number of epochs was 10 with a batch size of 20 and the same learning rate as in the previous experiment. Fig. 13 illustrates the results of this experiment based on accuracy, batch size and number of epochs when the dropout rate is 0.2.



**Fig. 13** The accuracy of the model using a different number of epochs and different batch sizes where dropout is 0.2.

#### 4- Conclusion

This research contributes to the development of a key enabler for Industry 4.0—big data with an AI decision making tool. The proposed method can be used to optimise manufacturing processes in a wide variety of industries. The successful implementation of this research in Canadian manufacturing facilities will positively impact the realisation of cost savings in these facilities and the development of feasible and sustainable green systems. This would significantly increase the leverage of the competitive advantages of the Canadian manufacturing sector.

The process begins with training the model based on the acquired data about acceptable and defective products from images from the production line. The data is pre-processed and normalised to improve accuracy and minimize the loss. The trained data is transferred to the Compute-Canada Cedar supercomputer and real-time synchronised with the production station. The production line captures a picture of each item and sends it to the server for classification using the already trained machine-learning system. When an image is classified as defective, the server commands the production station to take the corresponding action, such as sending the item to the denials station. The integrated model has a high level of performance and achieving up to 97% accuracy in the detection of defective items. The model is implementing the missing I4.0 enables in iFactory facility by building a robust real-time decision making model, based on machine learning, CPS, and IoT.

#### Acknowledgement

This work is supported in part by the Natural Sciences and Engineering Research Council of Canada. Computation for this work was done in part on the supercomputer Cedar maintained by WestGrid and Compute Canada.

## References

- [1] Lu, Y., "Cyber physical system (CPS)-based industry 4.0: a survey." *Journal of Industrial Integration and Management* 2(03): 1750014, 2017.
- [2] Lu, Y., "Industry 4.0: A survey on technologies, applications and open research issues." *Journal of Industrial Information Integration* 6: pp. 1-10, 2017.
- [3] Xu, L. D., E. L. Xu and L. Li (2018). "Industry 4.0: state of the art and future trends." *International Journal of Production Research* 56(8): pp. 2941-2962,2018.
- [4] Lee, J., E. Lapira, B. Bagheri and H.-a. Kao, "Recent advances and trends in predictive manufacturing systems in big data environment." *Manufacturing Letters* 1(1): pp. 38-41, 2013.
- [5] Helu, M., D. Libes, J. Lubell, K. Lyons and K. C. Morris, Enabling smart manufacturing technologies for decision-making support. ASME 2016 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers, 2016.
- [6] Li, X., D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai and S. Wang , "A review of industrial wireless networks in the context of industry 4.0." *Wireless networks* 23(1): pp. 23-41, 2017.
- [7] Chen, C. P. and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data." *Information Sciences* 275: 314-347, 2014.
- [8] NIST, N. I. o. S. a. T., "Data Analytics for Smart Manufacturing Systems."
- [9] Narayanan, A. N., R. Ak, Y.-T. T. Lee, R. Ghosh and S. Rachuri , Summary of the symposium on data analytics for advanced manufacturing, 2017.
- [10] Babiceanu, R. F. and R. Seker, "Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook." *Computers in Industry* 81, pp. 128-137, 2016.
- [11] Moghaddam, M. and S. Y. Nof, "Collaborative service-component integration in cloud manufacturing." *International Journal of Production Research* 56(1-2): pp. 677-691, 2018.
- [12] Lv, C., X. Hu, A. Sangiovanni-Vincentelli, Y. Li, C. M. Martinez and D. Cao , "Driving-Style-Based Codesign Optimization of an Automated Electric Vehicle: A Cyber-Physical System Approach." *IEEE Transactions on Industrial Electronics* 66(4): pp. 2965-2975, 2019.
- [13] Gölzer, P., P. Cato and M. Amberg, Data Processing Requirements of Industry 4.0-Use Cases for Big Data Applications. Proceedings of the European Conference on Information Systems (ECIS) 2015.
- [14] ElMaraghy, H., T. AlGeddawy, A. Azab and W. ElMaraghy, Change in manufacturing–research and industrial challenges. Enabling manufacturing competitiveness and economic sustainability, Springer: pp. 2-9, 2012.
- [15] kaggle. , "ResNet-50." from <https://www.kaggle.com/keras/resnet50> (Retrieved Jan 2019).
- [16] Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.
- [17] EigenInnovations. "Making Complex Quality Inspection Simple (Visited Jan 20, 2019)." from <http://eigeninnovations.com/>, 2019.
- [18] FreeCodeCamp, "Structure of Convolutional Neural Networks (CNN) (Visited Feb 10, 2019)." Retrieved Feb, 2019, 2019, from <https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050>, 2019.
- [19] Hahnloser, R. H. R., R. Sarpeshkar, M. A. Mahowald, R. J. Douglas and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." *Nature* 405: 947, 2000
- [20] Boltzmann, L., "Studien über das Gleichgewicht der lebenden Kraft." *Wissenschaftliche Abhandlungen* 1: pp. 49-96, 1868.