Improve Operations of Real-Time Image Classification Utilizing Machine Learning and Knowledge Evolution

David Prairie
Electrical and Computer Engineering Dept.
University of Massachusetts Dartmouth
Dartmouth, Massachusetts United States
Email: dprairie@umassd.edu

Paul Fortier
Electrical and Computer Engineering Dept.
University of Massachusetts Dartmouth
Dartmouth, Massachusetts United States
Email: pfortier@umassd.edu

***Abstract*—This paper delves into the generation and use of image classification models in a real-time environment utilizing machine learning. The ImageAI framework is used to generate a list of models from a set of training images and also for classifying new images using the generated models. Through this paper, previous research projects and industry programs are analyzed for design and operation. The basic implementation results in models that classify new images correctly the majority of the time with a high level of confidence. However, almost a quarter of the time the models classify images incorrectly. This paper attempts to improve the classification accuracy and improve the operational efficiency of the overall system as well.**

***Index Terms*—Machine Learning, Tensor Flow, Image Classification**

I. INTRODUCTION

Presented in this research paper is a new and novel approach to machine learning design that maximizes the balance between accuracy, efficiency, solution justification, and rule evolution. This research attempts to improve four factors of machine learning within a single design. Different components of the system design shall aim to improve one of the four factors compared to traditional designs. During this research, traditional open source machine learning methodologies are altered to improve different aspects of machine learning, tested against one application. These traditional methodologies will be integrated using ensemble methods for enhancing the performance along with providing justifications for the classification results. This paper will discuss the preliminary results, data used, the analysis, and validation methodologies used.

During this research the Identifiable Professionals (IdenProf) Dataset was used for training and validating models. The dataset contains ten image sets of distinguishable professions, each set containing 900 images for training and 200 images for validation per profession. The expected outcome of this research is a two part system capable of analyzing a real-time feed and perform profession classification based on a remotely generated model.

This paper is broken up into a Background section, where high-level aspects of machine learning and knowledge-bases will be discussed. The Methodology will be discussed following the Background section. Following the Methodology section the preliminary results will be discussed in the Data Analytics and Results sections. Finally the conclusion will discuss planned future work and recommended further work.

II. BACKGROUND

Knowledge base systems enable problems to be quickly and accurately solved based on previous cases. Knowledge base systems also allow for problem solving of very complex situations at speeds humans alone would not be able to achieve at such accuracies. Throughout the last 20 years, knowledge bases have grown in applications to assist in everyday tasks. More recently, IBM's Watson, an Artificial intelligence supercomputer, is in the limelight through its uses in the medical arena and in personal taxes with H&R Block. Other applications of machine learning have been completed or are being developed include detecting insider threats, big data analytics, market analysis for proposals, condition-based maintenance, and diagnostics in the medical field.

In the medical industry, Watson has proven capable of making the same recommended treatment plans as doctors 99% of the time. Unlike traditional human doctors, Watson can use all available medical resources when making a patient's diagnosis. By having such vast amounts of knowledge, Watson can provide treatment options doctors may miss. Watson utilizes powerful algorithms and immense computing resources to analyze all medical relevant data to find "...treatment options human doctors missed in 30 percent of the cases." [2] Since Watson has so much computing power it is able to determine treatment plans for patients faster than human doctors could, allowing doctors to put patients on treatments faster with the intervention of Watson. Watson is one example of how knowledge base systems positively benefit society by efficient and accurate problem solving of complex problems. Additional research into knowledge bases will allow them to problem solve faster, with more accuracy, and with less compute power requirements.

Condition-Based Maintenance is the method of monitoring a system's components to determine what level of maintenance is required for the system to remain functioning. Using a Condition-Based Maintenance system for managing maintenance events allows professionals to be proactive with performing maintenance activities versus being reactive. Reactive maintenance involves replacing components after they already

 ISBN: 978-1-61208-741-2

failed, causing system downtime. When a system goes down for unscheduled maintenance or repairs, many different repercussions can occur depending on the affected system's role. Using air traffic control centers as an example, any unplanned downtime has the potential to disrupt hundreds of flights and cost a significant amount of money due to flight delays [11]. Machine learning can be used to assist professionals to determine optimal maintenance schedules while minimizing system down time.

Although some of the described applications may not be as drastic as life or death medical decisions, all still can greatly affect society. Utilizing machine learning allows organizations to detect threats, conduct predictive maintenance, and perform many repeatable decision-making tasks consistently and efficiently. By allowing a machine to learn over time through historical cases and building a knowledge base, the machine allows operators to make informed decisions by providing every available piece of information. Systems are able to make decisions in a fraction of the time compared to a human expert attempting to come to the same decision, however additional advancement is needed to make machine learning more accurate and efficient. Areas needing additional inquiries include indexing algorithms, storage solutions, and finally the decision-making algorithms themselves. Machine learning is important because of the wide range of applications and benefits provided through the decision making and predictions capable. As the field advances, machines will create predictions and perform decision making faster and more completely.

A. Deep Learning Frameworks - Tensor Flow

Tensor Flow is a deep learning framework built on the first generation framework called DistBelief. Both frameworks were developed by Google to advance technology for the public and for use in Google's wide range of data products [13]. One of TensorFlow's major improvements over DistBelief is its ability to scale up onto large distributed hardware platforms utilizing multiple CPUs and GPUs. Tensor Flow utilizes a master orchestrator to distribute work across the number of hardware platforms available, each individual platform then breaks the work down to be solved across each system's available CPUs and GPUs.

Benchmarks conducted by Google researchers showed the Tensor Flow framework performs, as well as other popular training libraries. However, Tensor Flow did not have the best performance statistics as other libraries in the study when tested on a single machine platform

B. Rule Evolution IB1 & IB2 Algorithms

The IB1 and IB2 algorithms are used to evolve a system's rules used for classification by incorporating new cases. The addition of more instances over time causes the machine to alter its rules to improve the probability of giving a correct prediction on future instances. Instances can either enforce existing rules or go against existing rules. Over the course of a training period, the IB1 algorithm will converge to the actual results based on altering its rules. IB1 requires data to have specific attributes, making cases distinct enough for the algorithm to learn over time. If the data does not have distinct attributes then the machine will not learn, since no strong points of comparison are available between cases [1].

A downside of the IB1 algorithm is the need to store all correct and incorrect classifications over the lifetime of the machine. The IB2 algorithm is a branch of the IB1 algorithm that does not require the storage of all classifications, only the incorrect classifications. The trade off of saving storage space is the increase in time required for the IB2 algorithm to learn to predict with strong accuracy [1].

During the evaluation of both the IB1 and IB2 algorithms, researchers determined both algorithms are able to achieve acceptable prediction accuracies in some situations. However, IB1 attains greater accuracies on each scenario when compared to the IB2 algorithm. The increase in accuracy for IB1 could be attributed to the storing of all classification events versus only the incorrect classifications.

III. METHODOLOGY/THEORY

This section discusses the methodology used in the completion of this research. The research processed followed the system engineering v-diagram during development. This section is further broken into a system architecture and software architecture sections.

A. System Architecture

The implementation, which is shown in Figure 1, is broken into four sections; a workstation computer, web server, raspberry pi, and a shared storage box. The workstation computer contains a NVidia GTX 980ti and is used for generating models based on the training images. Once the models are generated they are stored on a centralized shared storage array. The raspberry pi is used as the real-time image capture device and performs classifications utilizing the models.

The web server is the middle point between the workstation and the raspberry pi, by serving the models generated for the Pi to download. To enable future learning from real imaging, the Pi will upload classified images to the web server for the Desktop to use in future model generation.

B. Software Architecture

The following software packages and frameworks are used to generate the models for real-time image classifications and for classifying new images:

- Python 3.6
- Tensorflow-GPU
- Numpy
- SciPy
- OpenCV
- Pillow
- Matplotlib
- H5py

 ISBN: 978-1-61208-741-2

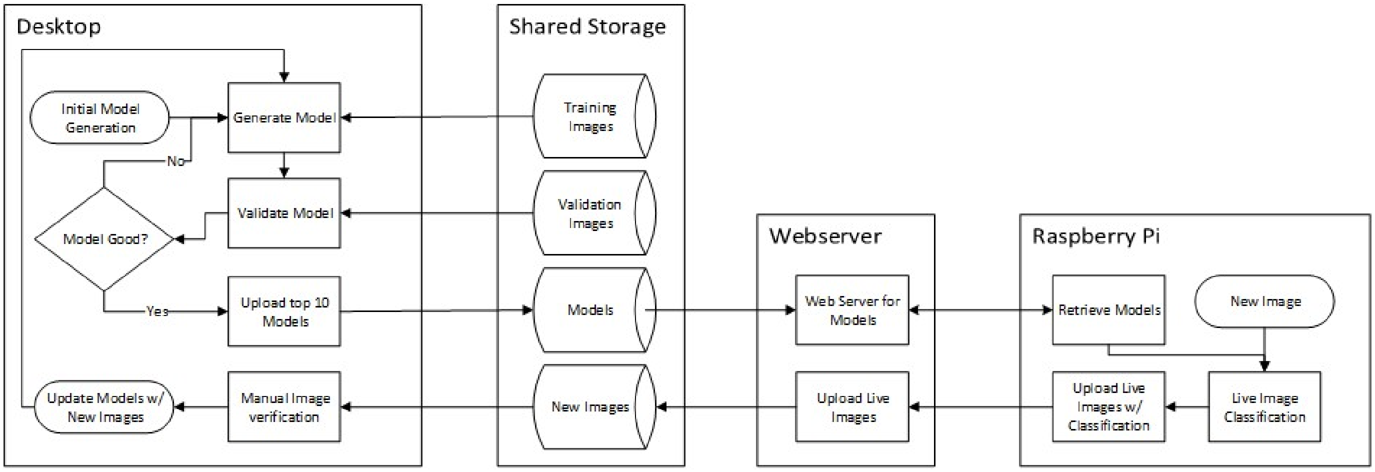

Fig. 1. System Flow Diagram.

- Keras
- ImageAI [10]

ImageAI is an API that can generate models based on an image set and perform image classifications based on the generated models. The API can generate models using the Desenet, Inveption v3, Resnet, and Squeezenet algorithms.

The workstation utilizes the above listed software when generating the initial models used for classification. ImageAI is a wrapper framework for the rest of the libraries, simplifying the development process. The same ImageAI framework is used on the raspberry pi for real-time image classification utilizing an add-on camera board. The raspberry pi is capable of handling the classification algorithms because the model generation and model evolution is offloaded to the workstation [9]. This heavily reduces the compute requirements, enabling the mobile real-time classification.

The web server is a repository for the raspberry pi to retrieve the latest generated models and to upload classified images for further analysis. Real-time images are uploaded to the webserver for manual verification of the image's classification and are then loaded into the desktop as additional training images to evolve the models. The combination of the Desktop and Raspberry Pi enables an overall system supporting model evolution while increasing the efficiency of the real-time classifier [9].

A future implementation adds a system capability of justifying the classifications provided. The current design behind the justification uses the built in confidence levels provided when classifying images. An alternative approach includes providing sample images that were classified using the same models and produced the same results.

IV. Data Analytics

During this research, the Identifiable Professionals (IdenProf) dataset [14] is used for evaluating the proposed changes to the ImageAI algorithm. IdenProf contains 10 distinguishable professions, listed in Table I. The dataset consists of over 900 images per profession used for training the system's models and an additional 200 images per profession for validating the models. All images are sized to a common pixel dimensions of 224 by 224 for uniformity. The image set has a make up of mostly white males from the top 15 most populated countries [14], compared to other genders or nationalities. During the duration of this research project additional images can be gathered by pulling images from Google's search engine.

TABLE I. Training Images Classifications

Training Images Classifications	
Profession	**Accuracy**
Chef	74.5%
Doctor	76.5%
Engineer	86.0%
Farmer	89.5%
Firefighter	90.5%
Judge	92.0%
Mechanic	84.5%
Pilot	87.5%
Police	87.5%
Waiter	72.0%

Current experiments include testing the base algorithms against the training and validation images. These 200 images will allow analysis and validation of the models generated at all three stages of development. Additional experiments will be planned utilizing the raspberry pi and camera for real-time image classification. The models used in classifications are selected based on the assigned accuracy defined during model generation. For these experiments the models select have over eighty percent accuracy.

Figure 2 depicts one of the test images for a pilot, one of the professions used in this research project. When running a classification against image, the system provides three results. Each result comes with a probability that the answer is correct. Typically the models generate one answer with a probability of over 95% and then the remaining two answers will make up the remaining percentage. In this case, the following probabilities are produced when classifying Figure 2: Pilot: 99.9527%, Chef: 0.0457%, and Mechanic: 0.0015%.

Fig. 2. Sample Profession Image - Pilot

V. Results

Initial runs for the system produced expected results, as shown in “Table I”. Two professions to take note of are the Chef and Waiter. From reviewing the Chef and Waiter images, there are many images where its difficult to distinguish which profession the image is of. Incorporating additional models for classification and using a voting scheme (described in the next section) could produce more accurate classifications.

When reviewing the model’s perceived accuracy on classification, the top profession is consistently listed as 95% or higher. This shows although the models are not able to reach the correct classification every time, the model produces a high level of certainty on the correct answer. When reviewing the raw data, a pattern arose showing when the models classified an image incorrectly, the probability produced tends to be significantly lower.

VI. Conclusion

At this stage in development the system has only been implemented using base algorithms and libraries with no other design improvements. The next step in development is to implement a voting scheme where the system will utilize multiple generated models to classify new images. Then the system will use the prediction from multiple models and use a voting method incorporating the model probability to determine a classification. The voting result is not necessary equal votes per model, the votes can be based on the model’s confidence in the answer the individual model is providing. For instance, if one model has a 98% confidence in the answer provided and the second model only has a 80% confidence, then chances are the first model would be the better of the two results to go with.

Upon completing the accuracy improvements, the system implementation will be broken up into the same format discussed in Figure 1. Currently the implementation has all model generation and model validation being conducted locally on Desktop. This is done to simplify the process, while ensuring the base algorithms are operating correctly. Once the remaining design are implemented a deeper system analysis will be completed. The analysis will include determining the reliability of the overall system and determining the theoretical versus actual efficiency of the algorithm.

Beyond the implementations and the analysis, additional work should be done collecting real-time images. The majority of the images used in this research project are collected from international researchers. A collection of images from additional countries, like the United States or United Kingdom, should be done to see if models are capable of classifying images correctly from other nations. This can also incorporate utilizing the raspberry pi at an event or walking around in pubic to do real-time classifications.

This research project attempts to incorporate common open source algorithms to enable real-time image classification using minimal processing power. By enabling the system to operate on minimal processing power at the user level, the system can be applied to new applications without relying on massive compute infrastructures at the endpoint. The base algorithms used will be modified to increase the operational efficiency and accuracy of the overall system.

The end result of this project can be applied to applications, such as security checkpoints or even used for field classifying insects and animals. As each component is implemented into the overall design, systems analysis for efficiency and reliability will be conducted to ensure the system is improving.

References

[1] D. Aha, D. Kibler and M. Albert, ”Instance-Based Learning Algorithms,” Machine Learning, vol. 6, no. 1, pp. 37-66, 1991

[2] D. Galeon, ”Paging Dr. Watson,” 28 October 2016. [Online]. Available: https://futurism.com/ibms-watson-ai-recommends-same-treatment-as-doctors-in-99-of-cancer-cases/. [Accessed 22 February 2019].

[3] Dtex Systems, ”The Hidden Security Threat,” Dtex Systems, 2016. [Online]. Available: https://dtexsystems.com/portfolio-items/infographic-findings-from-the-2016-costs-of-insider-threats-report/. [Accessed 21 March 2019].

[4] Q. Althebyan and B. Panda, ”A Knwoledge-Base Model for Insider Threat Prediction,” Proceedings of the 2007 IEEE Workshop on Information Assurance, vol. June, pp. 20-22, 2007.

[5] P. Bakkum and K. Skadron, ”Accelerating SQL Database Operations on a CPU with CUDA,” University of Virginia, Charlottesville, 2010.

[6] J. Jean, G. Dong, H. Zhang, X. Guo and B. Zhang, ”Query Processing with An FPGA Coprocessor Board,” in Proceedings of the International Conference on Engineering and Reconfigurable Systems and Algorithms, 2001.

[7] Martín Abadi et al, ”TensorFlow: A System for Large-Scale Machine Learning,” in 12th USENIX Symposium on Operating Systems Design, Savannah, 2016.

[8] D. Patil and P. Jayantrao. ”Malicious URLs Detection Using Decision Tree Classifiers and Majority Voting Technique.” Cybernetics and Information Technologies. 18. 11-29. 10.2478/cait-2018-0002.

[9] S. Jain, ”How to easily Detect Objects with Deep Learning on Raspberry Pi”, medium.com, Mar. 20, 2018. [Online]. Available: https://medium.com/nanonets/how-to-easily-detect-objects-with-deep-learning-on-raspberrypi-225f29635c74. [Accessed May. 11, 2019].

[10] DeepQuest AI, ”Official English Documentation for ImageAI!” DeepQuest AI. [Online]. Available: https://imageai.readthedocs.io/en/latest/. [Accessed: Feb. 11, 2019].

[11] C. Nichols, ”How many flights come in and out of LAX every day?,” Los Angeles Magazine, 1 May 2011. [Online]. Available: http://www.lamag.com/askchris/how-many-flights-come-in-and-out-of-lax-every1/. [Accessed Mar. 20, 2018].

[12] Homeland Security, ”Combating the Insider Threat,” Homeland Security, 2014.

[13] Google, ”Tensor Flow” Google, [Online]. Available: https://www.tensorflow.org/. [Accessed Apr. 20, 2019]

[14] M. Olafenwa, ”IdenProf Datasheet” Olafenwa, [Online]. Available: https://github.com/OlafenwaMoses. [Accessed Mar. 16, 2019]