

A Convolutional Neural Network for Automatic Identification and Classification of Fall Army Worm Moth

Francis Chulu¹, Jackson Phiri², Phillip O.Y. Nkunika³, Mayumbo Nyirenda⁴

Monica M.Kabemba⁵, Philemon H.Sohati⁶

Department of Computer Science^{1,2,4,5}

Department of Biological Sciences³

Department of Plant Science⁶

University of Zambia

Lusaka, Zambia

Abstract—To combat the problem caused by the Fall Army Worm in the country there is a need to come up with robust early warning and monitoring systems as the current manual system is labor intensive and time consuming. The automation of the identification and classification of the insect is one of the novel methods that can be undertaken. Therefore this paper presents the results of training a Convolutional Neural Network model using Google's Tensorflow Deep Learning Framework for the identification and classification of the Fall Army worm moth. Due to lack of enough training dataset and good computing power, we used transfer learning, which is the process of reusing a model trained on one task as a starting point for a model on a second task. Googles pre-trained InceptionV3 model was used as the underlying model. Data was collected from four sources namely the field, Lab setup, by crawling the internet and using Data Augmentation. We Present results of the best three trials in terms of training accuracy after several attempts to get the best metrics in terms of learning rate and training steps. The best model gave a prediction average accuracy of 82% and a 32% average prediction accuracy on false positives. The results shows that it is possible to automate the identification and classification of the Fall Army worm Moth using Convolutional Neural Networks.

Keywords—Augmentation; convolutional neural networks; classification; fall army worm; machine learning; tensorflow; transfer learning

I. INTRODUCTION

The Fall Army Worm possesses unique characteristics that make it a more devastating pest than other crop pests. It can spread quickly over a large geographical area, it can persist throughout the year meaning it is also able to affect irrigated plants and it can feed on over 80 different plant species [1][2][3]. The characteristics of the Fall Army worm enable it to cause massive damage to crops such as maize threatening the food security of countries where maize is the staple food. Its impact has been felt at national and household level [3]. Owing to its unique characteristics and the destructive nature of the FAW pest, the need to put up quick and proper monitoring systems of the insect can never be over emphasized. Currently the monitoring process of the insect being applied is the traditional pest monitoring process of manual observations and data collection and the use of Pheromone traps. The manual process is labor intensive, time

consuming, costly as it requires the engagement of field experts and requires one to count the insect for data collection which is likely to give false data [4][5]. Pheromone traps have proven to be effective in the monitoring of Pests in the field but still requires manual counting of the insect for data collection which is usually after weeks [6]. Delay in Pest data collection means a delay by would be stakeholders to apply appropriate counter measures to the Pest. In this regard, the use of computer based automated insect monitoring and classification is the solution [7] [8]. Machine learning is one of the technologies that are being applied in Agriculture in what is termed as Smart Farming to solve various challenges that the field faces. Its Application in Agriculture include but not limited to plant disease detection, farming area classification, plant recognition, fruit counting, Soil and vegetation mapping, Pest detection and management [9][10]. One of the problems in agriculture where machine learning can be applied also is the automation of Pest monitoring which requires in most cases the capture of images of live insects in the field. Deep learning is a recent branch of machine learning that constitutes modern techniques for image processing and recent studies show that it has successfully been applied in agriculture and has achieved unprecedented results [11]. Therefore in this paper we present the results of training a Convolutional Neural network model for identification and classification of Fall Army Worm. Due to lack of enough dataset and lack of computing power, we used data augmentation to supplement the dataset and transfer learning using the Google inceptionV3 model. The paper is divided into literature review, materials and methods, results, discussions and conclusion.

II. LITERATURE REVIEW

To supplement the results and get a broader understanding of the concepts of Deep learning and Convolutional Neural Networks, relevant literature was reviewed.

A. Convolutional Neural Network

Convolutional Neural Networks is the novel and most popular machine learning technology that has successfully been applied in computer vision and image classification. They are made up of a class of feedforward Artificial Neural Networks sharing weights [11]. The basic idea behind CNN is the local understanding of an image. They combine three architectural ideas, local understanding of the image, shared

weights and subsampling to ensure some degree of shift and distortion invariance [12]. The main architectural benefit of CNN is few parameters which in turn improve the learning time and a reduction in the amount of training data however enough training data increases the prediction accuracy. They are made up of Convolutional layers which act as feature extractors from input images, pooling layers which perform sampling operations on the dimensions of the input image thereby reducing the dimensions and fully connected layers which acts as classifiers by using the high level features learned [11][12][13][14]. Fig. 1 shows a typical Convolutional Neural Network Architecture where several convolutions are applied creating different representations of training datasets.

Traditionally classification problems were solved by first manually extracting image features and then serving the features as input to a trainable classifier which meant that the accuracy of the task was dependent on the extracted features. This limited the use of the manual approach [11][14]. Deep learning algorithms particularly Convolutional Neural Networks have proved to overcome the challenges of manual feature extraction but the increase in the complexity and size of the networks meant that a huge amount of training dataset is required which brings in the need for good computing power [15]. Training a good CNN from scratch requires a huge dataset and good computational resources. Fortunately one can use a technique called transfer learning to solve a classification problem. Transfer learning is the process of using a CNN that has been trained on a generic image classification task to solve a more specialized classification task. It is the use of a pre-trained model on another task as the starting point to another model of a different task. It falls into two categories: (i) Feature extraction which is the extraction of meaningful features from new samples by using representations learned by the pre-trained model in the previous work. This involves the training of a Top layer (classification layer) which is usually a Softmax function which is the output function of the last layer in neural networks whose purpose is to turn the score produced by the network into values that can be interpreted by humans. The Softmax function is defined as follows [15].

The function is given by: $\sigma : R^n \rightarrow \text{int}(\Delta^{n-1})$

$$\sigma(z) = \frac{\exp(\lambda Z_i)}{\sum_{j=1}^n \exp(\lambda Z_j)}, 1 \leq i \leq n \quad (1)$$

When $\lambda = 1$, (1) is called the standard Softmax function.

(ii) Fine tuning which is trying to unfreeze a few layers from the pre-trained model and training them together with the new classification layer [16][17][18]. Fig. 2 shows an example of transfer learning architecture.

There exists a number of pre-trained models which have successfully been used in recent years in transfer learning to solve classification problems such as ResNet50, inceptionV3, MobileNetV3 and VGG16. The models have been trained on huge datasets giving them the ability to generalize to new datasets when used in transfer learning [15]. Some of the models such as the InceptioV3 (Fig. 3) have been used in many image classification problems and have shown to have attained accuracies greater than 78.1% [19] [20].

The inceptionV3 was developed and trained using a Deep Learning framework called Tensorflow which provides several libraries for numerical computations [17][20][21][22]. In addition, Tensorflow provides a web based dashboard for visualizing the ongoing training called Tensorboard and makes it easy to deploy the trained model using what is called Tensorflow serving. In this research we are using the Feature extraction part of transfer learning since the inceptionV3 model has been trained on a large dataset enough for it to generalize to our datasets. of course Fine tuning will be the best option to get better and higher accuracies.

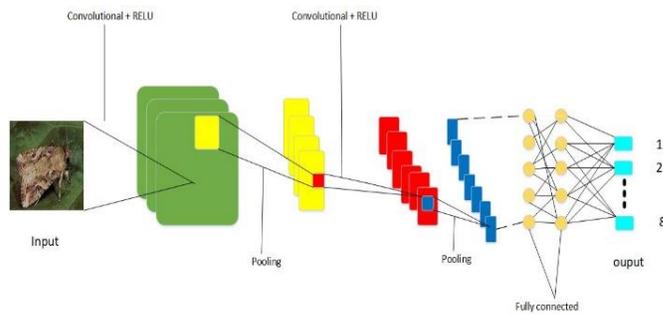


Fig. 1. Typical CNN Architecture.

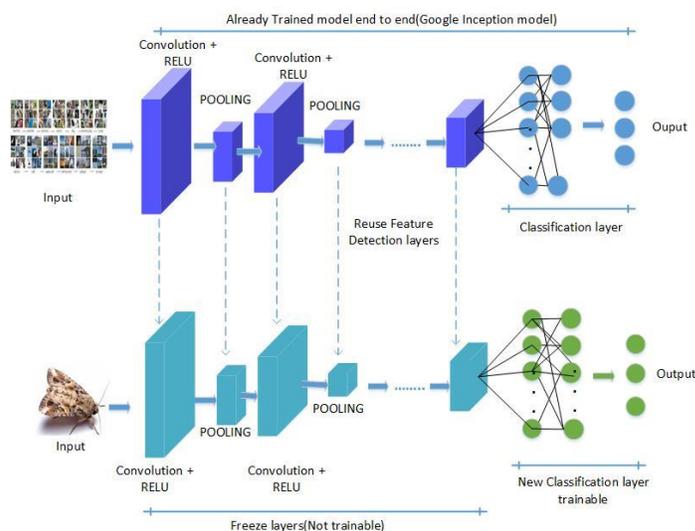


Fig. 2. Typical CNN Architecture.

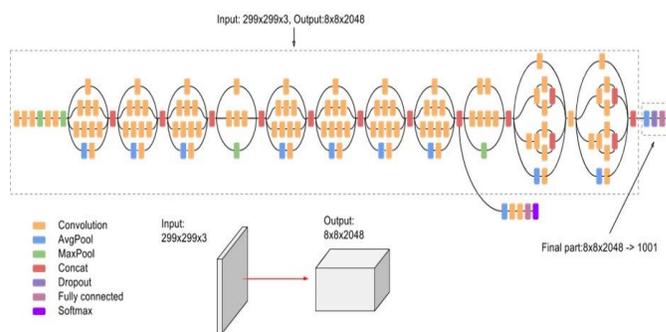


Fig. 3. InceptionV3 Architecture.

B. CNN Application in Classification Problems

The success that Convolutional Neural Networks have found in the classification problem domain can be attributed to a number of factors which may include but not limited to automatic feature detection, higher accuracies obtained, availability of good computing resources such as GPUs and availability of large datasets. Convolutional Neural Networks have successfully been used in many areas of agriculture in what is termed as Smart Farming. Some of the areas where CNNs have been used in agriculture include crop type classification, plant disease detection and recognition and so on [11]. In [18], the authors developed a model for predicting skin cancer cells by image classification using CNN and obtained more than 85% prediction accuracy and [23] tried to automate the classification of ovarian cancer using CNN to aid pathologists in cancer diagnosis. They used CNN based on AlexNet to predict ovarian cancer from cytological images. They obtained an accuracy of 78.20% by using augmented datasets. [24] Presented a self-operating stretcher developed using CNN transfer learning that solves the problem of transporting patients from long distances or short distances. The stretcher is trained to detect and identify the objects in its path. Authors in [17, 23, 24] showed that CNN are also applicable in medicine and are helping to solve many challenges that the field faces. [25] Applied CNN to the automatic classification of mosquitoes in the field. They used 4,056 mosquito images as training dataset and obtained classification accuracy of 82%. [26] Presented a method and the performance of the model for detection and classification of insects on sticky traps using CNN and obtained an average accuracy of 87.4%. Author in [27] applied CNNs in the identification of bacteria using 3D microscopy datasets. They tried to use CNNs to distinguish bacteria from non-bacterial objects from images obtained from zebrafish intestines using a microscopy. They deduced that CNNs are as accurate as human experts. The application of CNNs is broader and wide and results have shown that they are as accurate as they can be provided the correct training is done.

III. MATERIALS AND METHODS

A. Ethical Statement

The field data collection did not involve any species that is considered endangered hence no permits were required for us to collect data from the field.

B. Data Collection

Data was collected from 4 sources namely the field, Lab setup, and Internet crawling and data augmentation. Field data was collected from the Zambia Agriculture Research Institute research farms and taking photos of identified male FAW moth using a Cannon PowerShot SX430 IS Camera. With the aid of entomologists who are experts in FAW moth identification, we were able to obtain data from pheromone traps and from maize plants as FAW favors maize fields and most of them were resting in the fields as they are active during the night due to their delicate nature when exposed to the sun. In addition to taking photos and to supplement the collected data, we collected live FAW moths in beakers and set them up in a lab kind of environment. Since we will be dealing with the identification of live moths we tried as much as possible to take

photos of live moths strategically positioned in the pheromone trap or in the beaker though we also had images of dead FAW moths which made it easier to strategically position them. The collected data was not enough so we mixed the data from the field and lab setup with Internet images of FAW moths. We manually collected images from the internet by searching in the browser to make sure that the FAW moth images resemble the field images in terms of features and make sure that the collected image contains only the FAW moth as shown by Fig. 4.

The data collected from the field, Lab setup and Internet was not enough to give us confidence that we will train a reasonable model. To get a dataset that gave us comfort in terms of model training and testing, we used data augmentation, in fact 60% of the dataset came from augmentation. We used a python script which generated atleast 8 (67x65) more images as shown by Fig. 5 from an image by applying random cropping, random flipping, rotation using a rotation range of 40 degrees and random zoom.

Our focus was mainly on the automated identification of FAW moth but Convolutional neural networks requires that there are atleast two classes to perform a classification problem properly therefore we used other insects and moths which we referred to us false positives. There was no big reason that compiled us to use certain insects as the false positive class because it is hoped that the lure used in the traps which will act as the source of images will attract only FAW moth and majority of the insects attracted are FAW male moths. We therefore included a variety of insects with majority of them being moths which included but not limited to African bollworm moth, African Fly, Mosquitoes, Corn Borer moth, cutworm moths, stalkborer moths, stem borer moths and so on. The images of other insects were all collected by crawling the internet and we applied data augmentation to increase the dataset. We wanted to see how the model will perform if it was trained on unclear images (pixilated at 10pixels) and tested on clear actual FAW images, so we created another dataset of blurred FAW images from the dataset we made using a python script.



Fig. 4. Images Collected from the Field, Lab Setup and Internet Respectively.



Fig. 5. Sample of Image Augmentation with the First One being the Original.

C. Data Selection and Distribution

Once the dataset was structured, training, validation and testing images were determined. Before doing data augmentation, we selected 30 FAW images and 15 false positive images as test datasets. We selected mostly original clear unaugmented images as test images. The selected test images were not used in the training to avoid overfitting and were used only after the model was trained. We divided the test datasets into three parts of 15 images each namely Augmented, Actual Clear FAW and False Positives. The test Augmented images were a mixture of pixilated (at 5 pixels) images and randomly rotated, random vertically flipped and rescaled images. A total of 750 FAW images and 400 false positive images were generated using the process described in data collection section when data augmentation was done. Out of 750 FAW images, 10% were used as validation images since Tensorflow uses auto validation and the rest of the images were used as training dataset. From 400 false positives, 10% were used as validation dataset as well and the rest as training dataset. Table I shows the division of the dataset.

D. CNN Model Training and Testing

We used Transfer learning to train a Softmax layer on Top of the InceptionV3 model trained on the imageNet dataset which is a dataset of about 15 million labeled high resolution images with around 22, 000 categories [20]. The inception model gives high accuracy but is slower during training and looking at the dataset we had, it was the ideal choice. The training of the model was done on a Corei5 machine with 4GB RAM and 1TB hard disk. The operating system used was Ubuntu 18.04. We created a python virtual environment where all the dependents were installed to avoid conflicting dependent versions. Before training, Convolutional Neural Network algorithms demands defining parameters before training as they are likely to influence the classification results therefore we left all the parameters such as dropout, batch size, image crop and so on as they are defined in the pre-trained model and only altered the learning rate and training steps. We ran 8 trials and in each trial changing the learning rate and the number of training steps (epochs) in order to get the best and tested the models. From the 8 trials we got the best two trials which gave the best test accuracy and error rate. Out of the two selected trials, we picked the parameters of the model that gave the best results after being tested on test data and trained a pixilated FAW images model. The purpose of the pixilated trial was to test how the model will perform if trained on pixilated (blurred) images and tested on clear, augmented and false positive images. after training the Pixilated FAW images model, we ran the test data using the model and recorded the results and at the end we had three trials, namely, i) Clear FAW images, 0.01 learning rate, and 1500 training steps; ii) Clear FAW images, 0.001 learning rate and 5500 training steps; and iii) Pixilated (blurred) FAW images at 10 pixels, 0.01 learning rate and 1500 training steps and ranked them according to the best performing model in terms of identifying the actual and augmented FAW images and false positives.

TABLE. I. DATA DISTRIBUTION

Category	Total	Training	Validation	Test
FAW	780	645	75	30
False positives	400	360	40	15

IV. RESULTS

After a couple of trials to try and get the best learning rate and training steps, we finally got two of the best learning rates, 0.01 using 1500 training steps and 0.001 using 5500 training steps. We used 0.01 learning rate with 1500 steps to train another model using pixilated FAW moth images just to test the performance of the model when tested with clear images. We have grouped the results of the three tests into groups according to the model that performed better on all categories of test data.

A. Clear Faw Images, 0.01 Learning Rate, and 1500 Training Steps

Classification accuracies of round one model on all the three test categories are given by Table II. The model gave an average classification accuracy of 82% on actual clear FAW images, an average of 70.84% on augmented images and an average accuracy of 32.1% on false positives. The prediction accuracy was low in augmented images due to the fact that the images were distorted with most of them being pixilated. The model performed well in false positives which were images of other insects not FAW and were not used in the training dataset. Fig. 6 and 7 shows the Tensorboard training graphs and Fig. 8 shows the prediction accuracies.

TABLE. II. ROUND 1 PREDICTION ACCURACIES

#	Augmented %	Actual %	False positives %
1	97.8	90.1	54.7
2	84.3	98.2	49.8
3	97.6	97.6	24.1
4	19	90.9	47.5
5	40	99.4	42.9
6	95.8	93.7	80
7	31.6	51.7	45
8	89.4	94.2	0.1
9	77.9	94.4	71.1
10	47	43.5	0.2
11	64.1	74.8	7.4
12	87.9	86.2	1.2
13	79.6	69.8	52.7
14	92.6	47.9	1.3
15	58	97.5	2.7

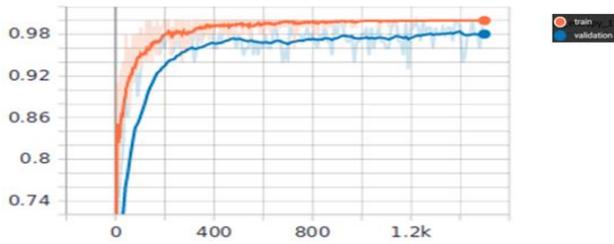


Fig. 6. Round 1 Training Accuracy.

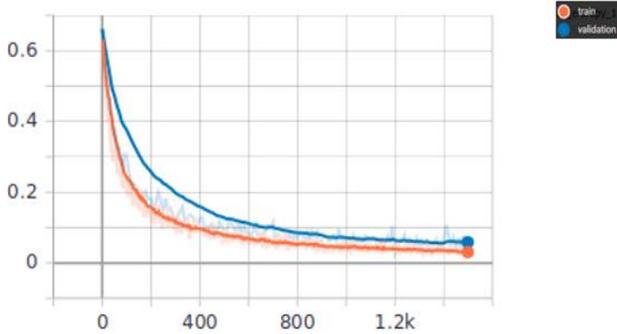


Fig. 7. Round 1 Cross Entropy.

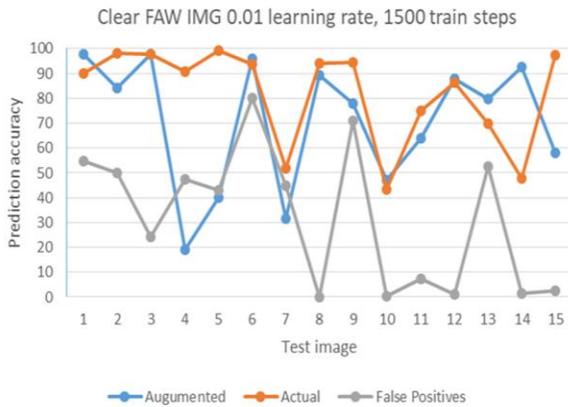


Fig. 8. Round 1 Prediction Accuracy.

TABLE. III. ROUND 2 PREDICTION ACCURACIES

#	Augmented %	Actual %	False positives %
1	95.8	90.1	5.2
2	80.9	96.3	33.8
3	95.2	67.5	20
4	12.2	74.2	19.1
5	41.1	91	4
6	86.5	91.7	2.9
7	73.5	50.3	93.4
8	77.5	89	46.8
9	80.9	98.7	94.5
10	85	17.9	47.8
11	77.5	89.2	94
12	89	95.7	56.7
13	93.6	39.3	73.5
14	95.2	50	87.4
15	61	96.7	53.1

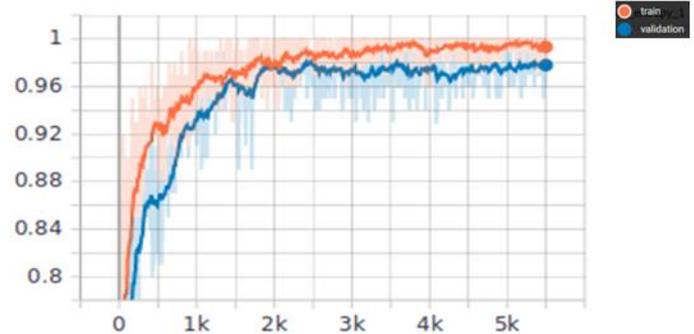


Fig. 9. Round 2 Training Accuracy.

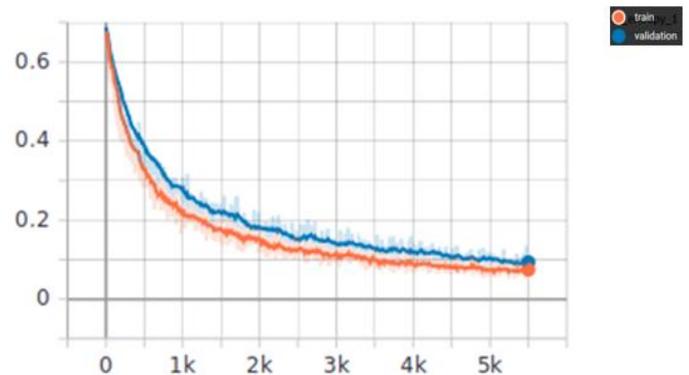


Fig. 10. Round 2 Cross Entropy.

B. Clear Faw Images, 0.001 Learning Rate, and 5500 Training Steps

Table III shows the prediction accuracies of the second round model and gave average prediction accuracies of 75.8% on actual clear FAW images, 76% on augmented images and 48.8% on false positives. The drop in the accuracies on actual images might be attributed to the rate at which the model was learning as we reduced the learning rate. Fig. 9 and 10 shows the Tensorboard training graphs and Fig. 11 shows the prediction accuracies.

C. Pixilated (Blurred) Faw Images at 10 Pixels, 0.01 Learning Rate and 1500 Training Steps

Table IV shows the prediction accuracies of the third round model. The model performed badly in all categories when tested on all three test categories as shown in Fig. 12.

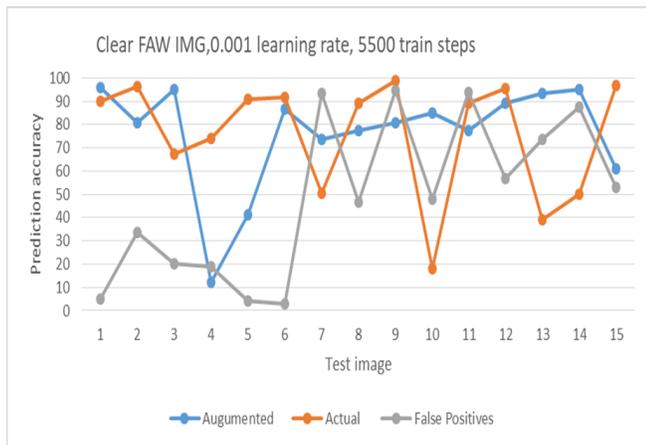


Fig. 11. Round 2 Prediction Accuracy.

TABLE IV. ROUND 3 PREDICTION ACCURACIES

#	Augmented %	Actual %	False positives %
1	94.3	17	93.2
2	3.9	7.8	99.7
3	66	1.6	99.6
4	8.6	62.6	1.2
5	13.3	9.2	1.9
6	99.1	28.5	9.7
7	12.6	11.2	93
8	25	1.2	53
9	9.6	1.3	14.8
10	53.9	16	42
11	99.8	9.4	100
12	40.7	0.3	100
13	6.5	11.8	97.5
14	39.3	12.6	0.5
15	51	14.8	0.7

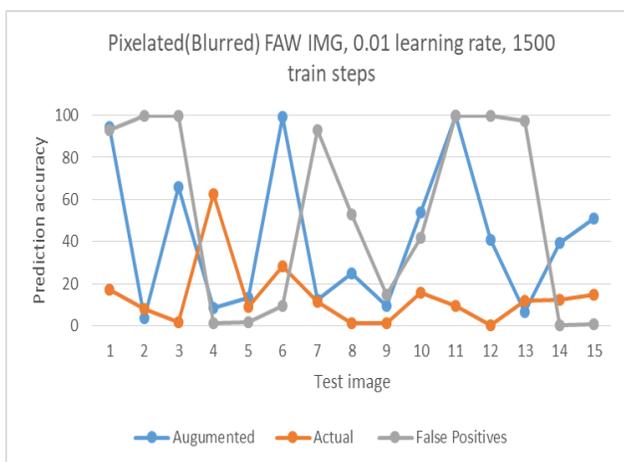


Fig. 12. Round 3 Prediction Accuracy.

V. DISCUSSION

We tested a model for the automatic identification and classification of Fall Army worm using convolutional neural networks. The major challenges we faced in the study included but not limited to; the size of test data which was fairly little and we had to rely on data augmentation which most likely affected results since to train a good CNN a good amount of data is needed. The lab setup was a bit challenging when dealing with live moth as it was near impossible to place them strategically hence we relied on the dead ones most of the time. The other challenge was the lack of previous work on Fall Army Worm using CNN which could have provided us with an opportunity to build on what others have done. Lack of good computing power meant that we had to use a pretrained model instead of building our own from scratch which would have given us an opportunity to fine tune the network in a way that suits our needs and hence get better results.

As CNNs requires a fairly large amounts of data to train a good recognition model and we had little data, we applied transfer learning using the inceptionV3 pre-trained model. The prediction accuracies were largely affected by little training data. After several trials the results of round one gave a better prediction accuracy (82%) which could greatly be attributed to the higher learning rate. Many improvements can be done to try and improve the prediction accuracies such as using fine tuning instead of feature extraction transfer learning and unfreeze a few feature extraction layers of the inception model and train them together with our new classification layer. We could also try and increase the training data set by collecting more images from the field and the internet and applying augmentation even though augmentation does not guarantee that the model will learn any new features. The other improvements we could use is the removal of noise from the current dataset such as removing the entire background from images. Another factor to consider would be the introduction of other insect classes because in the presented results our focus was on FAW only hence we just introduced a false positive class and added insects which are likely to be attracted to the FAW pheromone hormone.

VI. CONCLUSION

In this paper we investigated the use of Convolutional Neural Networks to identify and classify Fall Army Worm moth. We have presented results of three models built using transfer learning and the best model gave an average prediction accuracy of 82%. This is an ongoing research and future work may include but not limited to; increasing and cleaning the dataset by continuing to collect more from the field and the internet, trying the feature extraction of transfer learning and use the extracted features in a support vector machine and see how the results compare with the results of this paper, apart from the inceptionV3 we intend on trying out other pre-trained models such as mobileNet and compare the results with that of inceptionV3. The other work that will attempt to do is the use of more advanced deep learning algorithms and see how the results compares. The results show that it is possible to build a CNN model for FAW classification hence we develop a monitoring tool based on the model to achieve near real-time monitoring of the Fall Army worm.

ACKNOWLEDGMENT

We would like to acknowledge the Food Agriculture Organization of the United Nations (FAO) for the funding of this research and Zambia Agriculture Research Institute (ZARI) for allowing us to conduct field visits and collect data in their research farms.

REFERENCES

- [1] F.Chulu, J.Phiri, P.Nkunika, M.Nyirenda, M. Kabemba, S.Chiwamba, "Developing an automatic identification and early warning and monitoring web based system of fall army worm based on machine learning in developing countries", *Zambia ICT Journal*, Vol. 3, pp. 13-20, March 2019.
- [2] F.Chulu, J.Phiri, P.Nkunika, M.Nyirenda, M. Kabemba, M. Moonga, "Developing an automated fall army worm (FAW) Identification and early warning and monitoring system based on ANN techniques", Researchgate, 2019.
- [3] J.E. Huesing, B.M. Prasanna, D.McGrath, P.Chinwada, P.Jepson, J.L. Capinera, "Fall Armyworm in Africa: A Guide for Integrated Pest Management," 1st ed. 2018.
- [4] P. Abrahams; M. Bateman; T. Beale; V. Clotey; M. Cock; Y. Colmenarez; N. Corniani; R. Day, R. Early, J. Godwin, J. Gomez, P. G. Moreno; S.T. Murphy; B.O. Mensah, N. Phiri, C. Pratt; G. Richards, S. Silvestri, A. Witt, "Fall Armyworm: Impacts and Implications for Africa, Evidence Note (2)," September 2018.
- [5] L. Granado, Otoniel, M. Rach, Miguel, Migallón, Hector, Malumbres, Manuel, B. Pina, Alberto & S. Martín, Juan, "Monitoring Pest Insect Traps by Means of Low-Power Image Sensor Technologies". *Sensors* (Basel, Switzerland), December 2012.
- [6] A.M. Shelton, F.R. Badenes-Perez, "Concepts and applications of trap cropping in pest management", *Annual Review of Entomology*, Vol. 51, pp. 285–308. February 2006.
- [7] D. Epsky, Nancy, Morrill, Wendell, Mankin, Richard, "Traps for Capturing Insects" in *Encyclopedia of Entomology*, 2008, Ch. 10, pp. 3887-3901.
- [8] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, "A survey on image-based insect classification", *Pattern Recognition*, Elsevier, Vol. 65, pp.273 – 284, 2017.
- [9] S. N. Asiah Hassan, N. N. S. Abdul Rahman, Z. Zaw Htike, S. Lei Win, "Vision Based Entomology: A Survey," *International Journal of Computer Science & Engineering Survey*, Vol. 5, pp. 19-32, January 2014.
- [10] K.G. Liakos, P. Busato, D. Moshou, S. Pearson, D. Bochtis, "Machine Learning in Agriculture: A Review", *Sensors*, vol. 18(8), pp. 2674, August 2018.
- [11] A. Kamilaris, F.X. Prenafeta-Boldu, "Deep Learning in Agriculture: A Survey", *Computers and Electronics in Agriculture*. Vol. 147, pp. 70-90, 2018.
- [12] A. Kamilaris, F.X. Prenafeta-Boldu, "A review of the use of convolutional neural networks in agriculture," *The Journal of Agricultural Science*, Vol. 156(3), pp. 312-322, April 2018.
- [13] Y. LeCun, Y. Bengio, "Convolutional networks for images, speech, and time series", In *Arbib MA (ed.), The Handbook of Brain Theory and Neural Networks*, Cambridge, MA, USA, MIT Press, pp. 255–258, 1995.
- [14] Liu, T. Liu, S. Fang, Y. Zhao, P. Wang, J. Zhang, "Implementation of Training Convolutional Neural Networks", *Computer Vision and Pattern Recognition*, Vol. 2, June 2015.
- [15] W. Rawat, Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", *Neural Computation*, the MIT Press Journal, Vol. 29(9), pp. 2352-2449, September 2017.
- [16] M. Valan, K. Makonyi, A. Maki, D. Vondráček, F. Ronquist, "Automated Taxonomic Identification of Insects with Expert-Level Accuracy Using Effective Feature Transfer from Convolutional Networks", *Systematic Biology*, March 2019.
- [17] B. Gao, L. Pavel, "On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning", 2017.
- [18] V.B. Sai, Rao G, R.G. Narasimha, M. Ramya, & S.Y. Sujana.T. AnuRadha, "Classification of skin cancer images using Tensorflow and inception v3", *International Journal of Engineering & Technology*, Vol. 7, pp. 717-721, March 2018.
- [19] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.
- [20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *Computer Vision and Pattern Recognition*, Vol. 3, 2016.
- [21] Google Cloud Platform, "Advanced Guide to Inception v3 on Cloud TPU", Available online: <https://cloud.google.com/tpu/docs/inception-v3-advanced>. Accessed June 20, 2019.
- [22] P. Goldsborough, "A Tour of Tensorflow", *CoRR-Machine Learning*, Vol. abs/1610.01178, 2016.
- [23] M. Wu1 , C. Yan1, H. Liu1,Q. Liu, "Automatic classification of ovarian cancer types from cytological images using deep convolutional neural networks", *Biosci. Rep.* 38, BSR20180289doi:10.1042/BSR20180289, 2018.
- [24] S. Talpur, N. Khoso, "Advanced Ambulatory Operating Stretcher Learned by Means of Convolutional Neural Network (CNN)", *Journal of Biomedical Engineering and Medical imaging*, Vol. 5(3), 2018.
- [25] Motta D, Santos AÁB, Winkler I, Machado BAS, Pereira DADI, "Application of convolutional neural networks for classification of adult mosquitoes in the field", *PLOS ONE*, Vol.14 (1), 2019.
- [26] A. Nieuwenhuizen, J. Hemming and H. Suh, "Detection and classification of insects on stick traps in a tomato crop using Faster RCNN", <http://edepot.wur.nl/463457>, Researchgate, Jan. 2018.
- [27] E.A. Hay, R. Parthasarathy, "Performance of convolutional neural networks for identification of bacteria in 3D microscopy datasets", *PLOS Computational Biology*, Vol. 14(12), December 2018.