Oliver Clements, ocl26@uclive.ac.nz, 46994657

Cats, Camera, Action

Oliver Clements, Richard Green, Department of Computer Science, University of Canterbury

Abstract - This paper proposes the method for identifying known cats from each other from inside an image. The aim was to also keep the solution light weight so it could also be used on embedded hardware. This would allow for technologies such as cat specific doors, or "smart cat bowls" that open only for the desired cat.

The solution utilizes the YOLOv8 (You Only Look Once) CNN (Convolutional Neural Network) object detector. The training dataset consisted of 85 images where two distinct cats were present and to be identified from each other. Exposure and saturation augmentations were done by converting the RGB colour space into HSV. In the HSV colour space, exposure and saturation could be modified which would simulate different lighting conditions. This increased the dataset to 430 images in total. Three different models were trained which included a dataset without the augmentations, without increased validation set, and with the increased validation set. All the models were trained with 75 epochs each.

Results showed that the best performing model included augmentation and the increased validation set. This had a F1 score of 89%, accuracy of 81.5% and a mean confidence score of 80%. This performed much better than models from prior research that used classical object detection techniques which could only detect cats in specific poses. They also had lower accuracy and F1 scores. It even performed better than other researchers that used a CNN. This is largely due to the larger dataset of images used.

Limitation of model included that it has only been tested on two individual cats of the same breed. While this does signify the performance of the model, it is entirely possible that this model or method is not as reliable when different cat breeds are present.

I. Introduction

Facial recognition has been widely explored upon around humans [1], but much less has been done with animals, in particular household pets. Of the work that has been done, the majority is with dogs [2]. Applying these deep learning and computer vision techniques for cats, could have applications such as, finding lost pets [3], in-depth online pet registration, cat specific doors or individualized cat bowls. This technology can then be adapted to other animals in industries such as agriculture or animal well fare.

The purpose of this paper is to allow a user to input images of their pets to some software which will then be used to train a convolution neuro network (CNN). This will allow the cameras to distinguish between the owner's pets and the neighbors. Ideally the solution would be able to be used on embedded hardware and use in real-time. This paper will develop the back-end software that is responsible for the identification of the individual cats.

II. Background

The Chinese University of Hong Kong [5] created an object detector capable of detecting the heads of cats using classical object detection techniques. The data set used were all of cats in a canonical pose to reduce variability of the target. This also meant that the images of the target had to be in a normalised rotation. The faces of the targets were identified using a joint shape and texture detection. This was done by using two sliding windows to detect the shape and the texture of the target. Then convolutions were used to blur the outside of the image leaving the eyes and nose unblurred. From there a shape detector was used to detected the nose and eyes. After this, confidence scores were given. This method had an average precision of 63.2% with some minor obstruction. The main disadvantages with this method is that all the targets need to be in a canonical pose and normalised in rotation. This would also only identify the head of the target.

Debate is had when it comes to using deep learning for facial recognition of animals due to the black box nature of this method which can make validating results challenging [6]. Other approaches have been used such as landmark recognition. This is done by annotating the key landmarks on the targets face to train a CNN. The paper also used the deep learning network, You Only Look Once (YOLO) v8, to undertake the annotation process automatically. It is mentioned that the annotation done by YOLOv8 led to the

CCN to detect the faces of cats better than the manual annotation method. It is worth mentioning that YOLO failed to annotate 2.3% of the data set where no bounding box was given to the target in the image. While YOLO failed to annotate the faces of some of these images, it performed much better when there were multiple targets, target was tilted at a significant angle, and when there was partial obstruction of the target.

Due to colour inconsistency, small datasets, and the low quality of some images, detection of animals has been challenging in the past. This can be compensated for using augmentation on the data. Okafor [7] had a similar problem when trying to detect different animals from the air using an unmanned aerial vehicle. The solution they proposed was to do colour augmentation on the images. Then another image of the same object but viewed at a different angle was added to the dataset. By feeding the CNN with these colour augmented images it increased the peak detection from 90% to 99.5%.

Lan [3] used a CNN to distinguish between individual cats. The intended purpose of this project was to use an object detector to aid in finding lost cats. First the images were preprocessed to remove noise and other artifacts. Unfortunately, this paper did not specify what filters were used to do this. YOLOv3 was used to extract the faces of cats by assigning a bounding box with an assigned probability of the chance that the cat was in the box. YOLO then uses a regression model to form the final bounding box with the image inside of it. A Batch Normalization inception was then used for the facial recognition of the individual cats. This project only had a 45.5% of a successful detection but this was largely due to the data set used. The images from the data set consisted of many images being taken from inside cat shelters where the light levels are low and noise levels are high. There are also very few images of each individual cat which limited the ability of the network to be trained.

III. Proposed Method

1. Background Analysis

The background research suggests that using techniques such as landmark recognition, joint shape and texture detection are not recommended. Joint shape and texture detection undertaken in [5] has many constraints on the initial rotation and the pose of the target. Its accuracy of target detection is also limited. The landmark recognition undertaken in [6] while successful for detection of cats, cannot distinguish between different individual targets. However, this did reveal that a deep learning network such as YOLO could be useful in the annotation process when training a CNN. The detection of individual cats using YOLOv3 in [3] does not have the required precision required for this objective at hand. However, the CNN will have a larger dataset of each individual target in this proposal. To increase the dataset further and to make the training network more robust, augmentations will be done on the dataset. These augmentations will simulate different lighting conditions. It is suspected that these changes in the dataset will greatly increase the accuracy of the methods used by Lan.

The dataset used contains 85 original images. In these images there are two different individuals to distinguish between as seen below in Figure 1**.** To decrease the time spent training the network, Google Collab [8] will be used in the training process. Google Collab offers GPUs which can carry out many more floating-point operations per second than normal CPUs. This will therefore train the network faster [9]. The rest of the software will be carried out on an Apple Silicon CPU using Visual Studio Code as an IDE. YOLOv8 [10] will be the CNN used.

Figure 1: Targets to Identify between. Maple (Target 0) (Top), Tabby (Target 1) (Bottom)

2. Image Annotation

Image annotation involves applying a bounding box around each image with meta data that provides the ground truth of what the object is. Traditionally this is done manually. However, to increase the efficiency of this process, an established CNN was used. This CNN would detect all instances of a “Cat” class and apply a bounding box around it. This applied a bounding box of all the cats in the dataset. The class of each bounding box can then change to match the individual/target manually. The advantage of annotating images this way is the accuracy and the speed as seen in [11].

3. Exposure Augmentation

Exposure augmentation is varying the brightness of an image. It simulates how much light is reaching the sensor inside the camera which in turn, simulates different lighting conditions. Exposure augmentation was used on this dataset as it contains images in indoor and outdoor environments.

To change the exposure of an image the following was done:

1. Colour Space Conversion
2. Pixel Value Scaling

Colour space conversion is changing the RGB (red, green and blue) colour space to the HSV (hue, saturation and value)

colour space. This is first done by normalizing the RGB colour space from values from 0 to 1 as seen below in Equation 1. The value is simply the largest RGB value in the pixel as seen in Equation 2. The saturation is found by the values divisible difference to the minimum RGB pixel value as seen in Equation 3. The hue can be calculated as seen in Equation 4 [12].

$$R' = \frac{R}{255} \quad G' = \frac{G}{255} \quad B' = \frac{B}{255} \tag{1}$$

$$V = \max(R', G', B') \tag{2}$$

$$S = \begin{Bmatrix} 0 & \text{if } V = 0 \\ \dfrac{V - \min(R', G', B')}{V} & V > 0 \end{Bmatrix} \tag{3}$$

$$H = \begin{Bmatrix} 0 & \text{if } S = 0 \\ 60 \times \dfrac{G' - B'}{V - \min(R', G', B')} & \text{if } S = R' \\ 60 \times \left(2 + \dfrac{B' - R'}{V - \min(R', G', B')}\right) & \text{if } S = G' \\ 60 \times \left(4 + \dfrac{R' - G'}{V - \min(R', G', B')}\right) & \text{if } S = B' \end{Bmatrix} \tag{4}$$

Pixel value scaling is changing the value of the pixels by some factor in the HSV colour space. For the dataset used it was changed by 10% in both directions. The last step is to then convert the HSV colour space back to the RGB colour space. This is involving algebraically rearranging Equations 1 – 4.

4. Saturation Augmentation

Saturation augmentation involves changing the intensity of the colours for each pixel of the image. This is useful when the colours vary between images. Saturation was altered as most of the images in the dataset were taken using a smartphone. These do postprocessing on the images altering many of the image's attributes. One of these attributes is the saturation. This was deemed important as one of the visual characteristics of a cat are the patterns and colours of their fur. To increase the CNNs ability to distinguish cats from one another emphasis will be removed from the colour. This should instead train the CNN to detect the fur patterns which are intrinsic to each individual/target.

Changing the saturation of the images involves a similar process to changing the exposure of an image. The RGB colour space is converted to HSV which can be done using Equations 1 - 4. To adjust the saturation of the image the saturation value of the pixel is changed. For the dataset, the saturation was changed by 15% in both directions. The last step is to then convert the HSV colour space back to the RGB colour space. This involves algebraically rearranging Equations 1 – 4. The initial dataset contains 86 images. The exposure and the saturation augmentations increased the dataset size to 430 images.

5. Training the CNN

The object detector used was YOLOv8 which is a single convolutional neural network. YOLO predicts the bounding boxes and the probabilities of the target in a single evaluation. This makes it faster than other two stage object detectors which therefore makes YOLO better for real-time applications [13]. To decrease the computational time for both training and predictions the nano version of YOLOv8 was used. This will yield an object detector better suited for embedded hardware.

An epoch is the number of times the dataset is passed through the CNN. For the proposed method a total of 75 epochs will be used to train the network.

For each epoch the CNN splits each image into a SxS grid. Convolutions are then done to each cell throughout the grid which yields two new results. Numerous bounding boxes are placed, each with an assigned probability that an object exists inside each cell. Next a class probability map is created where each cell is assigned a class based on what object the CNN "thinks" is inside each cell, if any exist. This information is then combined to yield the detections in the image [14]. This process is visualized below in Figure 2.

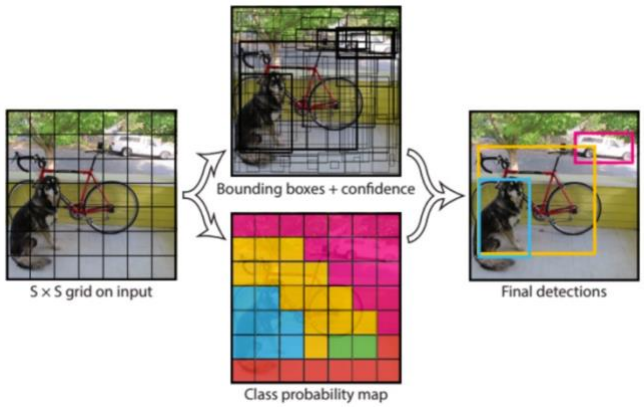

Figure 2: YOLOv8 Detection Diagram [14].

The training process of the CNN involves predicting where objects are in the image against the ground truth value. The ground truth value of each value is set during the annotation process which is simply information of where the bounding box is on the annotated image. The loss function describes how well the CNN has predicted where the objects are within the image. This is the mean square error between the prediction and the ground truth. The CNN then takes and uses this information to determine how well prediction model is performing. This process occurs within each epoch [15]. The images used in this stage of the training are known as the training images. Validation images are assessed with the training images but do not include the CNNs hyper parameters. Instead, validation images provide an unbiased review of the training model. This prevents the CNN from overfitting the data which occurs when the CNN associates' noise and other artifacts in the training images to be an attribute of the object [16].

The dataset was divided so that 85% of the images were used for the training set, and 15% were used in the validation set. No images were given to the testing set as this does not improve the model in any way. The images that are used for

testing are kept in a separate folder, so the augmentations do not get processed on them.

To increase the speed of the CNN, the images in the dataset were scaled down to 800 by 800 pixels. This is larger than the standard 640 pixels, as the details in the targets come from the different patterns in their fur. It is critical to not downsize the image so much that these details are removed. The images were downsized using Bicubic Interpolation as it retains more of the image data and colour than Nearest neighbor interpolation. This is important as discussed prior.

IV. Results

To evaluate the performance of the model, multiple metrics will be used to gauge how well each model performed. Accuracy is defined below in Equation 5 and measures how often the CNN makes a correct prediction. Precision is defined below in Equation 6 which measures how many predictions were correct out of all the predictions made. Recall is defined below in Equation 7 which measures the CNNs ability to detect instances of an object. F1 score is the mean between recall and precision. F1 is defined below in Equation 8. Any detections that had a confidence score less than 0.5 were discarded as they were deemed to be insufficient evidence for a successful detection.

$$\text{Accuracy} = \frac{\text{True Positives}}{\text{Testing Dataset}} \tag{5}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{7}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

The performance of all 4 iterations can be seen below in Table 1. All the models were evaluated against the same 26 testing images which were not included in the training or validation sets.

Table 1: Performance of all iterations

Model Version	Accuracy	Precision	Recall	F1
Without Augmentation	0.667	0.783	0.783	0.78
With Augmentation. No Validation	0.778	0.955	0.808	0.88
With Augmentation and validation set	0.815	0.917	0.88	0.89

Both the models with augmentation, and the one with the validation set had a similar F1 score performance metric with 0.88 and 0.89 respectively. The main differences in the models were their ability to detect objects in the dataset. The model without using validation images in the training process, had more false negatives where it failed to detect any object in the image. Whereas the model with the validation images in the training process, detected them as the incorrect object. The confusion matrices for these two models can be seen below in Figure 3 and Figure 4.

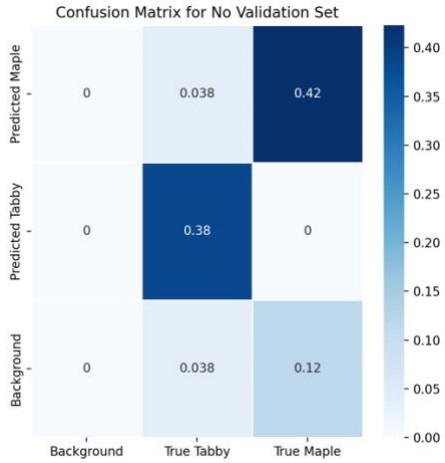

Figure 3: No Validation Model Confusion Matrix

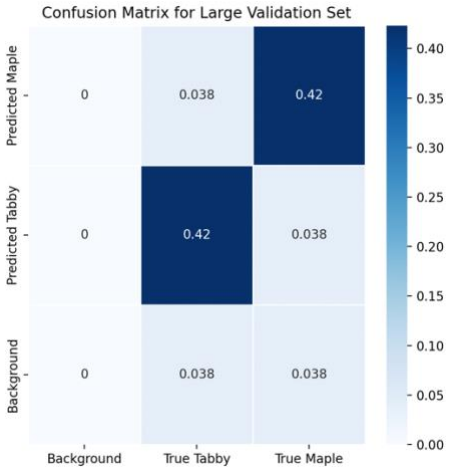

Figure 4: Validation Set Model Confusion Matrix

One of the key differences between the no validation set and the validation set models were the confident scores given. The mean confidence score for the no validation set model was 0.68. Whereas the mean confidence score for the large validation set was 0.80. This means the large validation set model has been trained better to identify each of the targets. So although the two models have a similar performance when measuring the metrics such as F1 scores, the large validation set model actually performs much better. Some detection examples from the large validation set can be seen in Figure 5, Figure 6 and Figure 7.

Figure 5: Detection of Tabby (1)

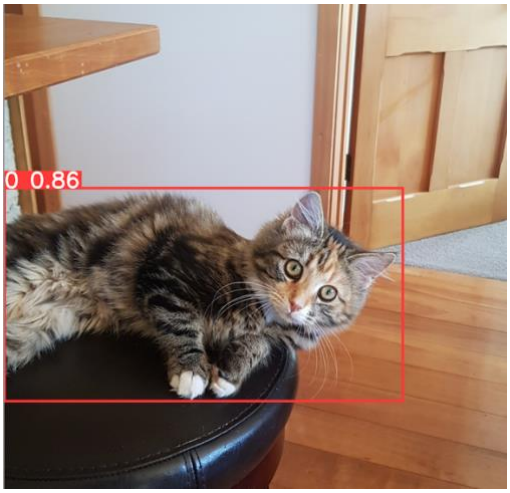

Figure 6: Detection of Maple (0) while laying down.

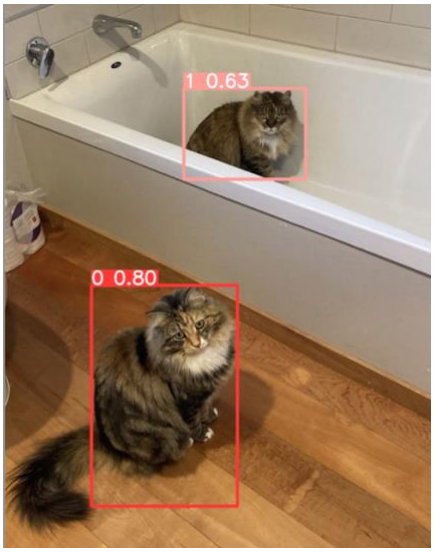

Figure 7: Detection of both Tabby (1) and Maple (0).

The results show that the proposed method was indeed successful at detecting different cats from each other. This was possible unlike the prior research such as Lans work [3] as a larger dataset was used in this paper. In this specific paper it proved that validating the results were straight forward due the CNN being successfully trained unlike in [6]. Using a CNN also allowed successful detection of targets being partly obscured or in a pose that in non-canonical position which were not possible in [5].

The model that performed the worst was the model without augmentation. While this model still possessed a mean precision, recall and F1 score of 0.783, its accuracy was only 66.67%. This disparity between the metrics such as precision and accuracy were due to the model failing to detect anything in the image. The reason for this is that the dataset this model was trained off was 4 times smaller than the models that trained of the augmented data. It is also worth noting that the mean confident scores of the true positives was 0.68. This provides further evidence that using a larger dataset to train a CNN makes the model more reliable than in [3].

Some of the limitations with the proposed method is the time taken to get the model operational. Significant amounts of time were taken in the annotation, training and validation stages. This can be costly when used in commercial environments. Also this paper only used two cats to distinguish between which were very similar. It is possible that this method may not work as effectively when deployed upon different species of cats which do not have stripy features.

V. Conclusion

The results show that the proposed method was successful at detecting different cats from each other using YOLOv8. The model that had the greatest success used augmentation to increase the size of the dataset and simulate varies conditions, and used validation in the training process. This model had an accuracy of 81.5% along a F1 score of 89%. This model also had a mean confidence score of 80%.

The proposed method outperformed the approaches found in prior research such as [3] which only had an accuracy of 45.5%. This was largely due to the dataset in the proposed method being larger and more augmentation techniques being used. The proposed method also outperformed the use of classical object detection techniques used in [5] which had an average precision of 63.2%. These could also only be used on cats in a canonical pose.

Limitations with the proposed method is the time taken to get the model operational. Significant amounts of time are taken in the annotation, training and validation stages. Also this paper only used two cats to distinguish between which were the same breed. It is possible that this method may not work as effectively when deployed upon different species of cats which do not have stripy features.

Future improvements could be made on this model. This would start out with testing the current model with many more species of cats to see it the results are translatable. To help remove the false negatives a two stage CNN approach could be used. This would first use YOLOv8 detect all cats in the image. With the bounding box data gathered it would then attempt to match each detected cat to a class in the CNN. This would involve a greater computing power to achieve but would increase the detections rates. Also accuracy could be improved by using a larger YOLO model such as a medium

or a large. These larger models are more powerful and use more convolutions on the data. These often lead to better detections. This is viable solution if the hardware used has significant amount of memory or real time is not a needed.

VI. References

[1] S. N, H. S and G. M, “Facial Recognition Using Deep Learning,” *Advances in Data Sciences, Security and Applications.,* vol. 612, pp. 375-382, 2019.

[2] M. Thierry Pinheiro, P. Mauricio Lisboa, W. Rafael de Oliveira and V. Eduardo, “Where Is My Puppy? Retrieving Lost Dogs by Facial Features,” *Multimedia Tools and Applications,* vol. 76, pp. 2-5, 2017.

[3] Y. Lan, “Pet Finding: Computer Vision Approaches for Pet Face Recognition and Verification,” UNIVERSITY OF CALIFORNIA, Los Angeles, 2022.

[4] H. Regan, “Facial Detection and Recognition in Feline Pets,” University of Canterbury, Chirstchurch, 2020.

[5] W. Zhang, J. Sun and X. Tang, “Cat Head Detection - How to Effectively Exploit Shape and Texture Features,” The Chinese University of Hong Kong, Hong Kong, 2008.

[6] G. Martvel, Shimshoni and A. Zamansky, “Automated Detection of Cat Facial Landmarks,” *International Journal of Computer Vision,* pp. 2-13, 2024.

[7] E. Okafor, L. Schomaker and M. A. Wiering, “An analysis of rotation matrix and colour constancy data augmentation in classifying images of animals,” *Journal of Information and Telecommunication,* vol. 2, no. 4, p. 465–491, 2018.

[8] Google, “Colaboratory,” Google, 2024. [Online]. Available: https://research.google.com/colaboratory/faq.html. [Accessed 18 May 2024].

[9] Y. Natsume, “Speeding Up Model Training with Google Colab,” Medium, 27 May 2022. [Online]. Available: https://medium.com/@natsunoyuki/speeding-up-model-training-with-google-colab-b1ad7c48573e. [Accessed 19 May 2024].

[10] G. Jocher, “Ultralytics YOLO,” Ultralytics, 10 01 2023. [Online]. Available: 'https://ultralytics.com'. [Accessed 16 May 2024].

[11] T. Tobia, C. Pasquale and B. Lamberto, “A CNN-RNN Framework for Image Annotation from Visual Cues and Social Network Metadata,” Quantexa Ltd, London, 2020.

[12] C. Vladimir, A. Jarmo and B. Vladimir, “Integer-based accurate conversion between RGB and HSV color spaces,” *Computers & Electrical Engineering,* vol. 46, pp. 328-337, 2015.

[13] E. Cengil, A. Çinar and M. Yildirim, “Cat-Dog Face Detector Based on YOLOv5,” *2021 International Conference on Innovation and Intelligence for Informatics,* pp. 149-152, 2021.

[14] M. Ahmed, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker and M. Z. Afzal, “Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments,” *Sensors,* vol. 21, 2021.

[15] Deepbean, Composer, *How YOLO Object Detection Works*. [Sound Recording]. Deepbean. 2023.

[16] R. Ruizendaal, “Deep Learning #3: More on CNNs & Handling Overfitting,” Towards Data Science, 13 May 2017. [Online]. Available: https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d. [Accessed 10 May 2024].