Action Recognition among Animals

Evaluating approaches and impact of Knowledge Graphs

Somansh Budhwar

2698242

s.budhwar@student.vu.nl

Bachelor Artificial Intelligence
A thesis presented to the Faculty of Science for
the Bachelor degree of Artificial Intelligence

Supervisor
Dr. Ilaria Tiddi

July 17 2022

“Research is formalized curiosity, it is poking and prying with a purpose.” -
Zora Neale Hurston

Contents

Abstract

Action recognition research using machine vision has evolved rapidly in the last decade. The human action recognition research has used various approaches to identify the humans in a video and classify their actions. Earlier approaches focused on object detection models, optical flow of videos. Recent approaches have integrated Zero-shot learning with Knowledge Graphs, and newer approaches are being proposed such as Disentangled Action Recognition with Knowledge-bases proposed by Luo. This is because Knowledge graphs allow extraction of new information than is already present in a scene. Video of a person dancing in America's Got Talent (AGT) will be interpreted as dancing by a simple model. However, a Knowledge Graph based model will recognise the AGT logo, and infer that this person is competing in a reality show.

However, action recognition among animals has not evolved as rapidly as human action recognition research, use of optical flow and Knowledge Graph is underutilised as an approach. This thesis seeks to further this area of research firstly by comparing two approaches of action recognition. One approach uses simple RGB Image frames of a video to determine the action, and the other approach uses its Optical flow to determine action. Secondly, to evaluate the effect of integrating Knowledge Graphs into the action recognition methods - using a custom dataset based on cats.

The first aim is to compare Convolutional Neural Networks (CNN) trained on standard RGB video frames with Optical flow of video frames to predict actions. Second aim is to evaluate the importance of integrating Knowledge Graphs and videos while predicting action recognition among animals. The actions to be recognised include: sleeping, walking, eating, scratching and jumping. The dataset includes 20 videos of different cats for each action. Using RGB images results in 20% accuracy, while optical flow images yield 26% accuracy. Optical flow images yields a better result, but the difference is only 6%.

In the second part, a video frame is analyzed to detect the cat, any toys, living beings, and food items near the cat. This information is used to infer further information from a Knowledge Graph. The RGB image is then combined with Knowledge Graph data about nearby objects to classify the action. It yields 43% accuracy. So, the improvement after integrating Knowledge Graph is almost double as compared to CNN trained on just RGB images. This means there is a significant advantage to be gained from using Knowledge Graphs in action classification.

However, further research is needed to back these conclusions since the dataset used in this thesis was small, and not of high quality. Another improvement needed is a better object detection model to detect more objects, and a larger Knowledge Graph which can provide better insights. The pre-trained models and Knowledge Graph used in this thesis are very basic. Nonetheless, it provides pathway to further research in the field.

1 Introduction

The digital revolution is not only beneficial to humans but also to millions of other species living on this planet. Today, wildlife sanctuaries, farms and pet owners generate huge amounts of video data for a diverse group of animals. This data is crucial for monitoring wildlife, analyzing social behaviours among animals, monitoring farm animals for physical and mental health, and conducting research. Moreover, with the changing climate and unsustainable practices, our oceans, forests, wetlands, and cryosphere are changing rapidly. This translates to a tectonic shift in animal behaviour which needs to be studied well. Since a large part of the data on wildlife exists in video format, scientists would need to sift through a large collection of videos. Then manually trim and annotate videos for analysis. However, it is not humanly possible to go through all the videos, let alone annotate them for research. Here artificial intelligence can contribute.

Today, it is easily possible to create deep learning models with more than 90% accuracy for recognising animals from images captured by camera traps [1]. This is especially true for common wildlife and domestic animals such as cats, dogs, tigers, deers and so on. However, action recognition among animals, accuracy varies from 40% to 90% accuracy, depending on the animal [2] [3] [4] [5].

Firstly, animal actions are very different from each other. A crocodile running is very different from a spider's run or an ostrich's run. Similarly, some actions do not apply to actions that are common to others. For instance, a snake cannot fly, an octopus doesn't follow a straight-forward motion. Moreover, even for similar class of animals, let's say quadrupeds, the motion is not always identical. A monkey's run is different from a Cheetah's run. So, the range of diversity means each family of species has to be dealt with differently. Here, a context about the animal and its surrounding might be helpful in determining their actions. For instance, a Knowledge Graph containing information about a horse will contain its number of legs, family of species, eating preferences, its range of actions, and so on. So, if a horse is captured in a video with its face near the grass, a deep learning model may reasonably predict that it is eating. However, a carnivore animal like tiger with its face touching the grass will not be eating, unless there is a dead animal on the grass. So, it is more complicated and requires more effort to determine animal actions as compared to human action recognition.

Secondly, there has been a lot of work in human action recognition, while research in animal action recognition has not been at the same level. Consequently, the datasets for research and pre-trained models are not easily available. So, one has to collect the data from personal sources and public sources such as YouTube, then the collected videos need to be processed and annotated manually. This consumes a lot of time and reduces the time for actual research and experimenting.

Thirdly, the major approaches in animal action recognition focus on the animal, and do not take into account the context in which the animal is. This contextual information can be implemented using Knowledge Graphs(KG), which are widely used across the industry by companies such as Google and Amazon, to infer new information from the available information. KGs of animals can contain information such as animal's typical behaviours, likes, dislikes, body features and so on. This can be very helpful in inferring the most likely action of the animal. For example, if an ostrich is seen above ground in an image, it can be inferred that it is jumping and not flying because Ostriches do not fly. However, to the best of author's knowledge, the application of Knowledge Graphs in animal action recognition is lacking.

Consequently, this thesis seeks to compare and augment the approaches in animal action recognition. Firstly, it compares two popular approaches in human action recognition in context of animals. The first approach determines the action by analysing an RGB image frame of a video, and the second approach calculates the optical flow of video frames and uses it to identify the action. So the question addressed is, do RGB image frames of a video yield better accuracy in action recognition than optical flow of video frames? Secondly, this thesis explores the effect of integrating Knowledge Graphs in action recognition. In other words, does integrating a Knowledge Graph to the input RGB frames improve the accuracy of the prediction?

The structure of this thesis is as follows. Firstly, the relevant concepts such as CNN, Optical Flow and Knowledge Graphs are explained. Next, the previous work on animal action recognition is discussed, along with the approaches and ideas. Then, the approach, the custom dataset and the proof-of-concept Knowledge Graph used in this thesis is discussed. Following that, the experimental setup and the results are discussed. Finally, the challenges and some suggestions for the future work are mentioned.

2 Preliminaries

2.1 CNN

Convolutional Neural Networks are primarily used to extract patterns from an image. The basic idea is to analyze an image by analyzing small areas of an image, and then extract information from it. For instance, if a CNN were to learn from pictures of cats and apples, it would quickly figure out that cat images have one or two triangles (ears) above a circular shape (the head). Similarly, the CNN would understand the eyes as a small circle within an eclipse. Thus, CNNs are helpful in extracting shapes and patterns from an image, which are essential while discerning actions. In Figure 2.1 a CNN takes an input of size 28x28 pixels, the image of a number, classifies the digit, after passing it through convolutional and pooling layers.

2.2 Optical Flow

Horn (1981) defines Optical flow as "the distribution of apparent velocities of movement of brightness pattern in an image". Which means that it can track

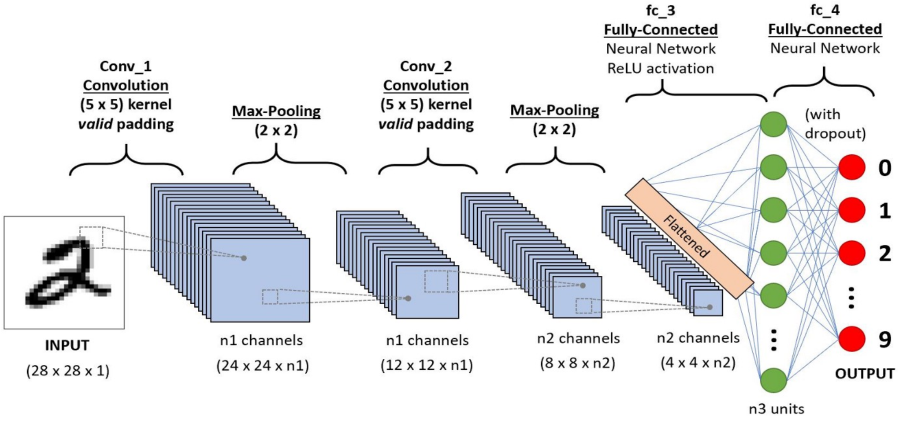


Figure 1: A Convolutional Neural Network.

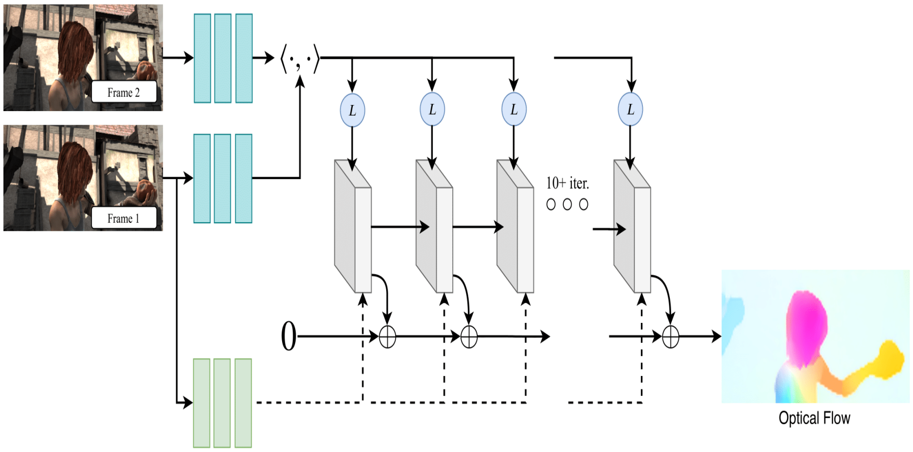


Figure 2: Princeton's RAFT (Recurrent All Pairs Field Transforms for Optical Flow) model architecture

the movement of an object based on its brightness in the scene. This in turn means that computers can trace the basic shape and movement of an object, and not be affected by the color, size or background of the object. For instance, if a cat is walking, the optical flow of its image will look almost the same as any other cat, regardless of the cat's color, size, and general background. This way of extracting body shape from an image is not only computationally cheaper, but also has the potential to improve accuracy of results in action recognition. This is why this method is popular in action recognition research. In this thesis, optical flow of the cats is calculated using Princeton's pre-trained RAFT model which is shown in Figure 2.2.

2.3 Knowledge Graphs

As per Krotsch and Weikum, a Knowledge Graph (KG) is “a network of entities, their semantic types, properties, and relationships” [6]. Each concept (node) is connected to another concept via a relation or a property (edge). For example, “Da Vinci” and “Mona Lisa” would be two nodes linked by an edge called “painted”. A sample Knowledge Graph can be seen in Figure 3. KGs allow us connect meanings and infer more knowledge from what we observe, an image. For example, if a system knows that Mona Lisa is a painting, it can use the graph in to gather knowledge about its creator and current location.

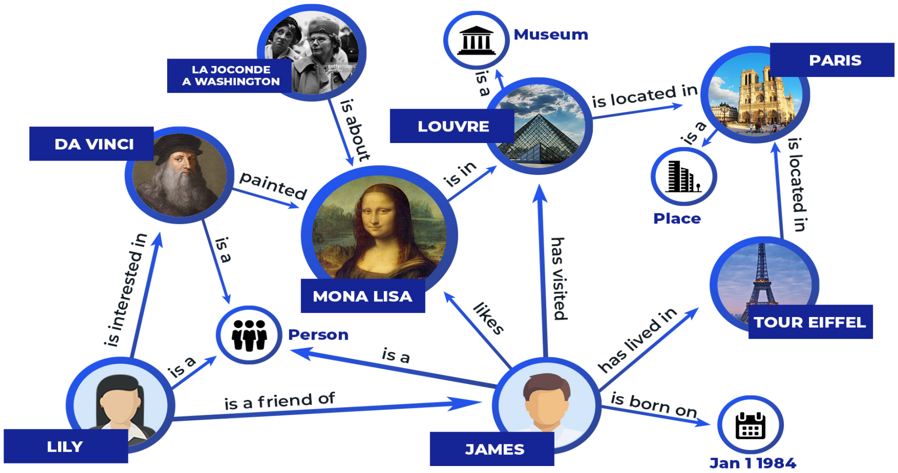


Figure 3: A sample Knowledge Graph containing linkages of concepts and properties of the painting “Mona Lisa”

There are multiple publicly available semantic networks online. ConceptNet is one such open-source semantic network that has a vast array of concepts and relations. It contains more than 34 million edges and contains concepts in more than 80 languages. For instance, the concept ’cat’ is linked to various other concepts such as its capabilities, needs, synonyms, body features and so on.

2.4 Cosine Similarity

$$\cos(\mathbf{A},\mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{1}$$

Cosine similarity (equation 1) measures the angle between two vectors, regardless of their length. It is widely used in text-mining while analyzing similarity between two documents. However, it is also a valuable loss function for training CNNs with a small dataset of images [7]. As Barz and Denzler have noted, a CNN trained from scratch does not yield good results, but using cosine similarity loss instead of cross entropy, it can achieve about 30% more accuracy. Since this thesis trains a model based on only 3000 images, cosine similarity was used as the loss function instead of cross entropy loss.

3 Previous work

Feng et al in 2021 used a spacial-temporal architecture to classify actions of wild felines [2]. They built an architecture with two parallel pipelines to analyze a video stream. One pipeline used the Mask Region-based Convolutional Neural network (R-CNN) [8] to outline the spatial features of the animal, and then passed it to a Tiny Visual Geometry Group (VGG) network for action recognition, thus classifying an action based on the static RGB image. The second pipeline used an LSTM network architecture called LEAP (Leap Estimate Animal Pose) to create a skeleton representation of the animal. It then analyzed temporal changes in these skeletal points to classify actions. They achieved 90% accuracy for 3 actions of wild felines galloping, ambling, and standing.

Schindler and Steinhage used Mask R-CNN to identify an animal from a video frame and then use that image to recognise the action [3]. They applied this method on wildlife animals such as deer, boar, fox and hare. The actions to be recognised were eating, moving and watching. Using this method, they achieved 63% accuracy in animal detection and 88% to 94% accuracy in action recognition.

Weining Li applied deep learning to recognise actions among wildlife animals [4]. The dataset included animals such as elephants, wolf, lion, chicken and a dozen others. The actions included rest, eat, walk and swim. The model used CNNs in two parallel streams, one detected the objects and the other detected generic features. However, the accuracy was low at 33% to 37%. Marcelo Feighelstein et al used convolutional neural networks (ResNet50) to detect pain in cats, by analyzing the facial expressions. Using image frames, and passing them through CNNs, an accuracy of 72% was achieved [5].

Action recognition among domestic animals such as mice has also been done. Hung and Chen used Raspberry pi-based YOLOv3-Tiny identification system to monitor domestic cats and recognise actions such as sleeping, eating, sitting down, walking, going to the toilet, and search on a trash can [9]. It used a pre-trained model for object detecting and after detecting the objects, the image was used to classify actions. They achieved 98% accuracy using this method. Geuther et al used static images from video frames to classify an action as grooming or not grooming. The architecture passed multiple video frames to a CNN and classify an action. Jingqiu used image entropy method for action recognition in cows [10]. The custom algorithm segments the cow's body parts in an image and then uses the segmentation shapes to determine the action such as hoof disease. However, most of the approaches do not take into account the surroundings of the animal. Moreover, many approaches focus on the static images of the animal, and ignore the temporal motion of the animal. This can lead to incorrect interpretations. For instance, it is not easy to discern between walking and standing action of an animal just by looking at one image. Similarly, the sleeping or sitting action of an animal could often look very similar to the eating action if we ignore food or water in the vicinity of the animal. Thus, there is a need to explore the importance of temporal and contextual data (using Knowledge Graphs) while predicting actions.

The most relevant work in the context of this paper was done by Nan Wu et al [11]. They combined Knowledge Graph(via ConceptNet) with CNNs to predict actions. The method used two parallel streams to process an image. In the first stream, the RGB image was processed via a CNN. Meanwhile, the pretrained model extracted the knowledge from the image, such as, whether it is a horse or has any stripes This information was thus embedded into a vector and used to train a model. The advantage is that it is not affected by factors such as light, color, background and so on, because the essence of an image is used to classify actions rather than the image itself. In the second parallel stream, the optical flow of the video frames was created and a separate prediction was used to compare the results. However, it is also not focused on animals. Moreover, it only notices the presence or absence of an object in the image, and ignores the aspect of distance among objects.

This thesis build further on the approach taken by Nan Wu et al, by integrating distance among objects in action recognition. Although this thesis applies action recognition approaches on one particular animals (cats), it can be used a generic framework for all animals.

4 Research Hypothesis and Questions

As we have seen, most of the current research involves focusing on the object and then using various architectures, most commonly CNNs, to recognise actions. Moreover, another common phenomena is to detect the shape of an object and then use that shape to determine which action was taken. Considering these approaches, this paper builds upon the present research and explores the following questions.

Firstly, can optical flow of video frames provide better accuracy over the simple RGB frames? Since it takes into account only the shape of the animal, it should be able to eliminate unnecessary information from an image while classifying actions. In other words, if the animal is reduced to its bare structure, and the variations in color, background, and lighting are removed, so would it yield a better prediction of their actions?

Secondly, can Knowledge Graphs improve the performance of current action recognition approaches? Thus, by providing structured knowledge about the context in which the animal is located, will the deep learning model achieve better results?

5 Approach

Firstly, “Cat” was selected as the animal of interest since cat videos are easily accessible from personal sources and social media. Then, 6 categories of actions were chosen such that these actions should be diverse, yet fairly common in the animal kingdom. Thus the actions chosen were, eating, sleeping, walking, jumping, scratching, and sitting. For the sake of simplicity, other closely related actions such as running, drinking, playing were avoided. Next, video data was collected for each action manually from YouTube and personal sources. From these videos, two datasets were prepared for the first research question. One dataset was just labelled data of RGB image frames extracted from the videos, and the other dataset was the Optical Flow of the video frames. Each dataset consists of images of size 1000x1000 pixels.

For the second research question, a third dataset was created. The idea was to identify objects in an image, and extract context from each image frame using a Knowledge Graph. The manual Knowledge Graph created for this paper is shown in the Figure 4. However, not every object in the image was identified, only objects relevant to the 6 actions were identified from the image. First set of objects were food items such as orange, bowl, banana etc. that can indicate the eating action of a cat. Second set of objects were living beings such as a dog or a human, because presence of a living being might influence the cat to walk or sit near that being. Third set of objects were Couch objects such as bed, sofa or chair, which might influence the cat to sleep or sit. Final set of objects were Toys such as balls, since a toy might influence the cat to chew it. So, a pre-trained PyTorch model was used to identify the objects of interest and its location relative to the cat was noted. Then, using the knowledge graph, the class of the object and the cat’s position relative to the object was inferred. For instance, if a bowl is located above the cat, using the Knowledge Graph, it can be inferred that there is a food item present near the cat, but it is directly above it. This can help the model deduce that the cat is probably not eating since the bowl is above it. Thus, KG helps in generating new information from an image and provides a clearer context to it. The RGB Image dataset was used to extract relevant information from the images and the information inferred using the KG is stored in a pandas dataframe with 12 columns. The details about dataset creation are mentioned in the section below.

Next, the models were created for both research questions. Following the common approach in image classification, Convolutional Neural Network was chosen as the model architecture to classify actions from images. The CNN architecture, as shown in Figure 2.1, was made up of two convolutional layers, a flattened layer, and two linear hidden layers of 800 and 400 neurons each, with 6 output neurons for each action class. It was designed to take images of size 300x300 pixel size. The input image was normalised and converted to a tensor before feeding it to the CNN. The RGB Image dataset was used to find the optimum hyper-parameters for the CNN such as convolution layer sizes, pooling technique and activation function.

Moreover, Cosine Similarity was used as loss function instead of Cross En-

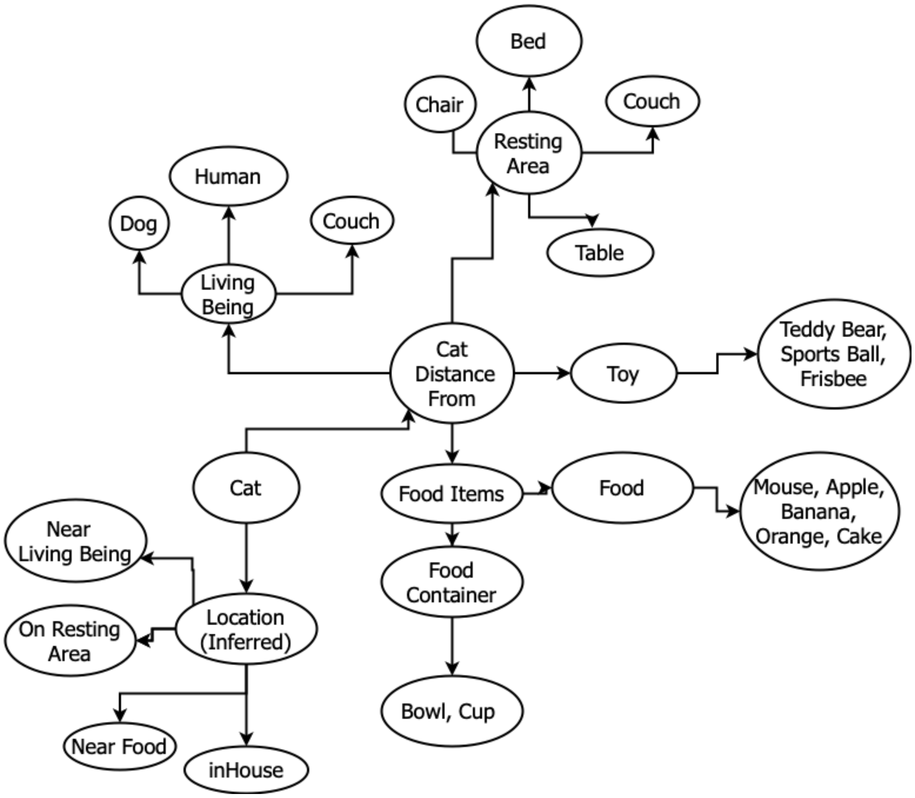


Figure 4: The manually created Knowledge Graph used in this paper.

tropy loss since it provides better accuracy for CNNs trained on small datasets [7]. Then, the optimised CNN architecture was used to train two models on the RGB dataset and Optical Flow dataset. Finally, the performance of both models were compared and analysed.

For the second research question, the existing CNN architecture was slightly modified (Figure 6) to integrate the knowledge stored in the pandas dataframe. The two convolutional layers were kept as it is, but in the flattened layer, 12 neurons were concatenated to integrate the data from the pandas dataframe. The hidden layers were also kept unchanged. The mixed input data of images and numerical data was used to train the customised CNN, and the results of this model were compared with the results of the model trained on RGB image dataset.

6 The Dataset

The dataset consists of about 20 cat videos for each action, namely: Jumping, Eating, Sitting, Sleeping, Scratching, and Walking. However, the number of

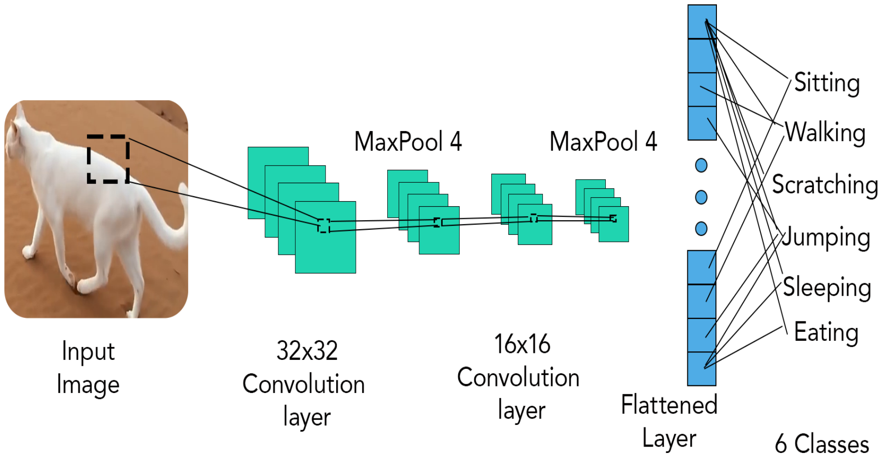


Figure 5: The CNN Architecture for processing RGB and Optical Flow image datasets.

videos for the category "jumping" are only 15. Some videos are shared in private capacity, thus cannot be shared publicly. However, most of the videos were downloaded from YouTube. The videos were manually trimmed to around 10 seconds each, and labelled after their category. The blurry frames or unclear videos were not kept in the final batch. But due to the nature of "jumping action", its contents were mostly blurry because the camera could not capture the fast motion clearly. The videos were stored in folders named after their category. Afterwards, a python script was run using cv2 library, which converted the video frames into png format images.

The frames obtained from each category of videos were stored in a separate folder named after that video inside their category folder. So, a video named "01scratch" would be stored in a folder called "scratching". The png format images of this video were stored in a separate location, but inside a similarly named folder "scratching". This creates the basic image dataset upon which three datasets needed for experimentation are built.

6.1 RGB Image Dataset

The first dataset consists of cropped images of cats. To identify cats, a pre-trained model "fasterrcnn resnet50 fpn" was used, which is provided in the PyTorch framework. If the model detected a cat with more than 60% confidence, a bounding box around that cat was drawn, cropped at that box, resized to

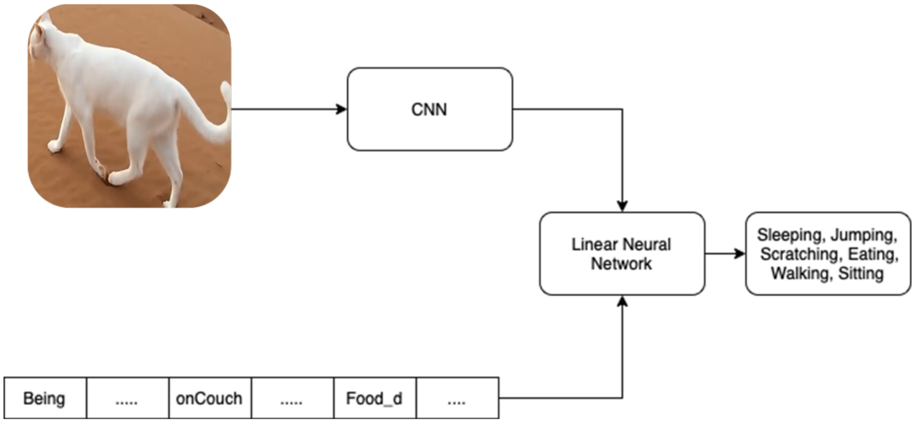


Figure 6: The CNN Architecture for processing RGB + KG dataset.

1000x1000 pixels, and the image was saved in its respective category folder. If the model does not identify a cat in the frame, the last known box location was chosen to crop the image. This was done because the cat maybe in motion and the image was blurry, especially relevant in jumping videos where cats were mostly blurry in many of the frames. In case there was no known last location, a center crop was taken. Since this was a time-consuming process, every 5th frame was processed, from the total set of frames resulting in approximately 3000 images. The result is shown in Table 2.

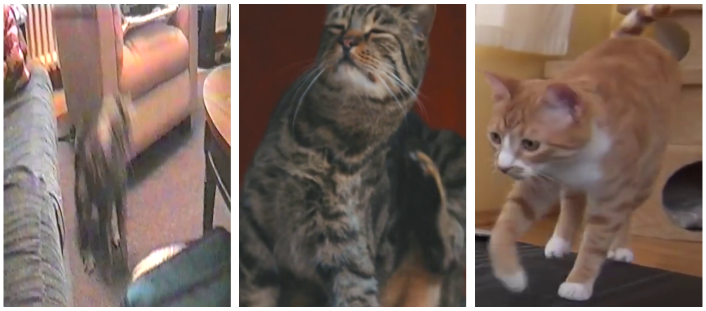


Figure 7: The cropped images of a cat jumping, scratching, and walking respectively (from left to right).

6.2 Optical Flow Dataset

In the second dataset, the base dataset images were processed using Princeton's pretrained RAFT model [12]. This model basically looked at two image frames, here 10 frames apart, and calculated the flow of pixel. So, for instance, the optical flow of a cat eating would yield an image see in Figure 6.2. Similar to the RGB Image dataset, every 5th image was processed, resulting in approximately 3000 images.

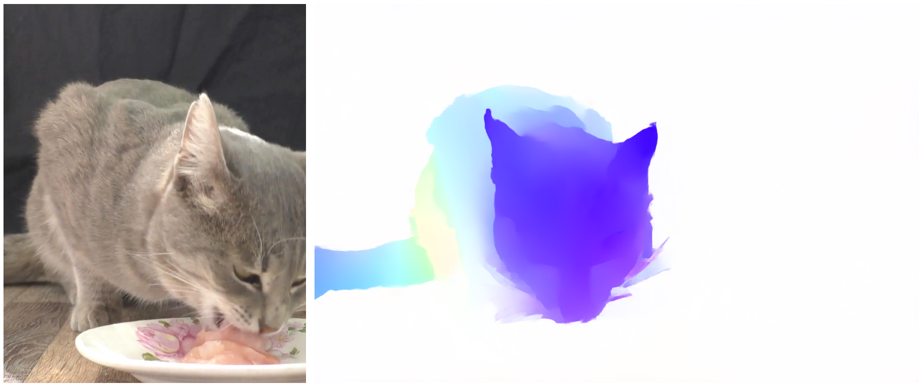
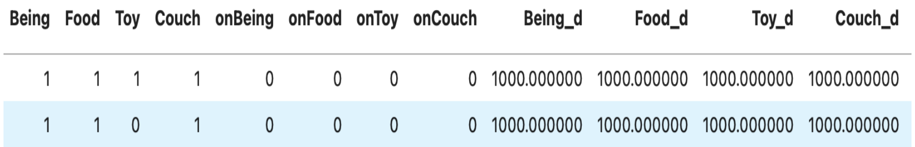


Figure 8: The optical flow image of a cat eating.

6.3 RGB + KG Dataset

In this thesis, a relatively simple KG (Fig 4) was created to infer knowledge about a cat's surroundings, by gathering basic information from the image. For instance, if the image consisted of a cat and a couch, such that the cat's location was above the couch and sufficiently close, we could infer that the cat was probably on the couch, and less likely to be eating food. Also, since cats do not eat dogs, if an image showed a cat was near a dog, it could be deduced that the cat was not eating. Thus, KGs can be essential in decoding a scene and help in recognising actions. Although a more elaborate KG is available at ConceptNet, it was not used because of the small scale of concepts needed for this paper. However, if the object detection capability is increased, ConceptNet KG should be a better fit.

Similar to the RGB dataset, the images from the base dataset were analyzed by the Resnet model. However, this time, not only the cat's location was noted, but also the locations of other objects of interest. The Resnet model can recognise about 50 different objects. In this thesis, the objects of interest were: person, bench, bird, dog, frisbee, sports ball, teddy bear, bottle, cup, bowl, mouse, banana, apple, orange, carrot, cake, chair, couch, bed, and cat. So, the Resnet model was used to capture information the above-mentioned objects while ignoring other objects such as train, skateboard, surfboard etc. The model automatically created a bounding box around the object detected. Consequently, the data was stored into a pandas dataframe. Since the Knowledge Graph used in this thesis is relatively simple, a python script was run on this dataframe to construct relevant knowledge as shown in Figure 9.



Being	Food	Toy	Couch	onBeing	onFood	onToy	onCouch	Being_d	Food_d	Toy_d	Couch_d
1	1	1	1	0	0	0	0	1000.000000	1000.000000	1000.000000	1000.000000
1	1	0	1	0	0	0	0	1000.000000	1000.000000	1000.000000	1000.000000

Figure 9: The Dataframe holding information derived from image and the Knowledge Graph.

The first 4 columns denote the presence or absence of an object by 1 and 0 respectively. So, in the first row, Toy has a value of 1 which means there is a toy present in the scene - or at least the model detects there is a toy. The next 4 columns denote if the cat is on top of the mentioned object. Thus, if onCouch is 0, it means that the cat is not located above the couch. This is calculated by comparing the bounding boxes for the couch and the cat. If the bounding box of the cat is above, but not too far above, it is considered that the cat is "on" a couch. The last 4 columns contain information about the distance of an object from the cat. This distance is calculated in pixels, and if the object is too far away or is absent, its value is 1000, because it is not likely to play a role in the action. It must be mentioned that calculating distance based on pixel distance is not ideal, because a couch may be far away from the cat, but from the camera's perspective it might look nearby. A solution to this can be to use relative size of object while calculating their distance. However, this approach would require an advanced Knowledge Graph, and thus it is beyond the scope of this thesis. The inferred data is stored in a csv file, and location of each frame analyzed is also stored within this file. A custom dataset is created in PyTorch to load this dataset.

7 Experimental Setup

The experiment consists of two parts. First, compare the accuracy of a CNN architecture using RGB Image dataset as input, versus using Optical Flow images as the input. Second, compare the accuracy of CNN trained using the RGB Image Dataset with the accuracy of CNN trained using RGB + KG dataset. Both parts have identical settings and hyperparameters, to avoid any confounding variables. So, hyperparameter tuning was performed on the RGB Image dataset for 5 epochs, and the parameters with minimum loss were chosen for all the experiments, as seen in Table 1.

The optimum setting for the CNN architecture, as seen in Table 2, is as follows. Each input image is resized and center cropped to 300x300 pixels, and

Conv Layer I	Conv Layer II	Learning Rate	Pooling	Loss
32	8	0.001	LPPool	1.1926
64	16	0.001	LPPool	1.0009
64	32	0.001	LPPool	1.3449
64	16	0.001	LPPool	1.3461
64	16	0.001	MaxPool	1.3588
32	16	0.001	MaxPool	0.7246
32	8	0.001	MaxPool	1.6120

Table 1: Hyperparameter tuning results.

transformed to a normalised tensor with mean value 0.5 and standard deviation of 0.225. Then it is passed to the first Convolutional Layer of size 32x32 without any padding, and stride 1. The layers are then passed to ReLU activation function, then pooled using MaxPool2d, which chooses the maximum valued pixel in a pool of 4 pixels. These layers again pass through the second Convolutional Layer of size 16x16 pixels. Again the layers are passed via ReLU and then MaxPooled. Then, the layers are flattened into a single layer of neurons, which is further connected to 2 hidden layers of 800 and 400 neurons each, which culminate into a final layer of 6 neurons which represent the predicted class of the image.

The architecture as for the CNN using Knowledge Graph, as seen in Figure 6, is identical to the CNN trained on RGB Images dataset, except that it concatenates 12 neurons to the flattened neuron layer. Thus allowing the information from the Knowledge Graph to integrate with the data from convolutional layers.

In each experiment, training and test data is shuffled and split in 80:20 ratio. Training is performed for 10 epochs for each CNN. Finally, the model is tested on the test set only once and the results are noted.

8 Results and Discussion

The accuracy for RGB images was 20% accuracy on test set, while the the accuracy of optical flow images was higher, at 26%. Considering there are 6 classes, a random classification would yield the result of 18% accuracy, because the training and test data was evenly distributed. However, it must be noted that the data has class imbalance because "jumping" class is only 5% of the total data, and "scratching" class is about 10%. So, the accuracy obtained is better than a random prediction, but also not much better. However, when the static images were combined with the data from Knowledge graph, the accuracy went up to 43%. So, adding information and context from the Knowledge Graphs improved the accuracy substantially. Thus, it is evident that KGs can add value to action recognition. Now let us take a deeper look into the results, as shown in Table 2.

Models						
Category	RGB		Optical Flow		RGB+KG	
	Precision	Recall	Precision	Recall	Precision	Recall
Eating	0.0	0.0	0.0	0.0	0.711	0.797
Jumping	0.0	0.0	0.0	0.0	0.0	0.0
Scratching	0.0	0.0	0.0	0.0	0.0	0.0
Sitting	0.185	0.561	0.219	0.975	0.518	0.888
Sleeping	0.0	0.0	0.0	0.0	0.0	0.0
Walking	0.251	0.527	0.718	0.534	0.211	0.432

Table 2: Performance of models for each category

8.1 RGB and Optic Flow Images

First thing to notice is that this CNN only predicts two classes, namely "sitting" and "walking". Of the 619 test images, it classifies 226 as walking and 339 as sitting. Thus precision and recall for every other class is zero. This model is only marginally better than a model that predicts only one class, and thus achieves 100% recall. The precision achieved for each class is not satisfactory because it is on average 22% for both the classes.

The CNN trained on Optical flow images yielded better accuracy overall, but it also predicted only 2 classes. Of 596 test images, 81 were predicted as walking, and 515 as sitting. This results in 97% recall for sitting, because the model is well tuned to classify "walking", and marks everything else as "sitting". The Optical Flow CNN has the edge over the RGB Image CNN because of its better performance in predicting the walking class with 70% precision. This results in better overall accuracy. Again, it must be mentioned that the dataset it too small, and this result may also be due to chance.

8.2 Static Images combined with Knowledge Graph

This approach resulted in prediction of 3 classes, namely walking, sitting and eating. It predicted 176 inputs as "eating", 213 as "sitting" and 212 as "walking". So, it yields significant improvement over the previous approaches and almost doubles the accuracy to around 43% as compared to RGB Images CNN. Another interesting result is that it predicts 3 classes, unlike just 2 in the above models. Furthermore, the precision and recall of "eating" class are particularly well, which is most likely due to the information about food in the KG data. Its precision and recall are also close to each other, which means that not only does the model identify eating precisely, it identifies eating scenarios more often. This improvement in predicting eating provides significant boost to this model over the simpler one. Thus, it is evident that the KG played an important role in evaluating the scene around the animal. This model also fares better than previous ones in predicting the sitting class. However, the performance of this model while classifying "walking" is marginally worse. The high values of recall for "walking" class in the first model may be because it only classifies two classes, so the chances of getting them correct is high.

The high precision of “walking” class in the optical flow images can be attributed to the clear images in case of walking cats. Since there is significant motion during walking, it is easier to calculate the optical flow as compared to stable actions such as sleeping. Looking at the dataset, it was clear that Optical flow images of “sleeping” was generally blurry. Finally, none of the models predicted “jumping” class for any image, which is most likely due to the blurriness and fuzziness of the images in all the datasets.

9 Challenges and Future Work

9.1 Lack of annotated data

Since the annotated video data on cats was not available, it was a time-consuming effort to search, trim, and annotate the data. A small portion of the data was personal videos, while most of it was sourced from YouTube. Since each category had 20 videos (mostly 15 seconds each) the model only had limited data to learn from. Usually, action recognition research papers use around 100 videos per action.

9.2 Quality of Data

Some concerns regarding the dataset are as follows. The number of videos per action are small, since it is a time consuming task to download, trim and label videos manually by only one person. Secondly, some videos have poor lighting, camera shakes and low resolution which reduces the overall quality of video frames. Hence, it impacts the overall results in this paper. Stable camera, better lighting, and high definition videos can definitely improve the accuracy of the models.

9.3 Motion of camera

In some cases, the camera recording the video itself was moving, leading to blurry images. This affects the image frames and leads to an incorrect depiction of the optical flow. This distortion of data further reduces the quality of the model. In fact, when the blurry images were removed from the Optical Flow images, the accuracy went up to 35% and the model predicted 4 classes, instead of just two. Though the blurry images were not discarded in this thesis, future works may consider using a larger dataset of high quality images.

9.4 Limited Computing Power

Due to the limitation of using only one GPU for this project, it took about 5 to 6 hours for one training session consisting of 5 epochs. Moreover, processing the images and creating the optical flow images was also time-consuming.

Consequently, only 20% of the video frames were used to create the training dataset.

Also, a limited set of objects were detected in a scene using the PyTorch Resnet model. This yields a limited knowledge about the scene, for instance, food items such as fish and meat are not detected, thus limiting the knowledge that there is food present near the cat, which limits the ability to predict that a cat is eating. For example, in "eating" videos, there was always a food item present in the scene. However, 13% of the time the model could not identify the food items. This was mostly in cases where the cat was eating catfood from a bowl. Moreover, the accuracy of the identified objects is also not good. For example, in "jumpig" videos, there was almost never any food item in the scence. But the model identified a food item 35% of the time in scenes. Future works may include pre-trained models provided by Nvidea which can detect thousands on objects in a scene, but that would also require more computing power.

9.5 Proof-of-concept Knowledge Graph

Rudimentary use of the concept of a Knowledge Graph, and must to be expanded further for better analysis. For example, presently, the distance between objects is calculated using pixel distance in the image. However, in cases where an object is near the camera (such as food) and the cat is far behind, but in the image they seem right next to each other, the model will infer them as being close-by. Using relative sizes of the objects can be used to properly calculate their relative distances. A Knowledge Graph that contains average size of subjects (cat, bowl, couch) can be used to calculate a more accurate relative distance among objects.

10 Conclusions

In this thesis, action recognition among animals was explored. First of all, data was manually collected from various sources, including personal and public sources. It was then trimmed, and annotated to create three datasets of RGB Images, Optical Flow Images, and RGB Images with data integrated from a custom Knowledge Graph. Next, the accuracy of CNN trained on RGB images was compared with the CNN trained on Optical Flow of video frames. Secondly, the role of Knowledge Graph was evaluated in improving results of action recognition.

Although optical flow images yielded better results, it is difficult to draw any conclusion from the results because the improvement is marginal, and as the dataset is too small to confidently comment on the performance of these approaches. However, it is evident that Knowledge Graphs can offer a significant improvement in classifying animal actions by introducing contextual knowledge. Moreover, considering the difficulty of gathering, and cleaning the data manually, it would be wise to focus on gathering and annotating more data on animals.

References

[1] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. E5716–E5725, 2018. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1719367115

[2] L. Feng, Y. Zhao, Y. Sun, W. Zhao, and J. Tang, "Action recognition using a spatial-temporal network for wild felines," *Animals*, vol. 11, no. 2, 2021. [Online]. Available: https://www.mdpi.com/2076-2615/11/2/485

[3] F. Schindler and V. Steinhage, "Identification of animals and recognition of their actions in wildlife videos using deep learning techniques," *Ecological Informatics*, vol. 61, p. 101215, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121000066

[4] W. Li, S. Swetha, and D. M. Shah, "Wildlife action recognition using deep learning," 2020.

[5] M. Feighelstein, I. Shimshoni, L. R. Finka, S. P. L. Luna, D. S. Mills, and A. Zamansky, "Automated recognition of pain in cats," *Scientific Reports*, vol. 12, no. 1, p. 9575, Jun 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-13348-1

[6] M. Krötzsch and G. Weikum, *Journal of Web Semantics*, vol. 37-38, pp. 53–54, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570826816300026

[7] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," 2019. [Online]. Available: https://arxiv.org/abs/1901.09054

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[9] R.-C. Chen, V. S. Saravanarajan, and H.-T. Hung, "Monitoring the behaviours of pet cat based on yolo model and raspberry pi," *International Journal of Applied Science and Engineering*, vol. 18, pp. 1–12, September 2021. [Online]. Available: https://doi.org/10.6703/IJASE.202109$_1$8(5).016

[10] G. Jingqiu, W. Zhihai, G. Rong-hua, and W. Hua-rui, "Cow behavior recognition based on image analysis and activities," *International Journal of Agricultural and Biological Engineering*, vol. 10, pp. 165–174, 2017.

[11] N. Wu and K. Kawamoto, “Zero-shot action recognition with three-stream graph convolutional networks,” *Sensors*, vol. 21, no. 11, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3793

[12] Z. Teed and J. Deng, “RAFT: recurrent all-pairs field transforms for optical flow,” *CoRR*, vol. abs/2003.12039, 2020. [Online]. Available: https://arxiv.org/abs/2003.12039