The 19th Electronics New Zealand Conference (ENZCon), Dunedin, New Zealand, 10-12 December, 2012

Real Time Monitoring of Cat Feeding Behaviour

Said Al Souti*, Donald Bailey*, and David Thomas^

* School of Engineering and Advanced Technology (SEAT)
^ Centre for Feline Nutrition (CFN)
Massey University
Private Bag 11 222, Palmerston North, New Zealand
al_sooti933@hotmail.com, D.G.Bailey@massey.ac.nz, and D.G.Thomas@massey.ac.nz

Abstract: Cat food manufacturers currently compete intensively against each other to gain and maintain market share. They are striving to produce a product that is palatable, healthy and preferred by the majority of the cats, by understanding more about cats' feeding behaviour. Therefore, in this paper, we propose an efficient and economically viable method to monitor cat feeding behaviour. The proposed method was based on a background subtraction algorithm to detect the cats if they were in the feeding area and trigger the video recording accordingly. Background subtraction was performed by calculating the absolute difference of the background and the new frame. Then the result was thresholded using Otsu's method to extract the cat's body from the background noise.

The proposed method was implemented and tested for one week in the Centre for Feline Nutrition, a total of 20 hours of monitoring, and it showed an extremely accurate performance, where a single i5 Processor was able to monitor and record videos of 4 cages at 25-30 fps simultaneously. In addition, the algorithm was able to provide a successful detection rate of 94%, with a 5% false acceptance rate and a 1% false rejection rate.

Keywords: Monitoring cats, running average background subtraction, template matching, cross correlation, load cell.

1 INTRODUCTION

Cats are New Zealand's most popular pet with 48% of households owning an average of two cats. For most of these cats, commercially prepared foods form the sole source of nutrition. Therefore, consumers expect high standards of quality, nutrition and value for their pets. To achieve this standard of quality, pet food manufacturers are continually working to increase their knowledge of pet nutrition, ensuring the food is palatable and healthy [1].

The Centre for Feline Nutrition (CFN) has the responsibility of assessing and quantifying the quality of different cat food products and ultimately deciding which formulation is preferred by the majority of a test panel of cats. Their current testing procedure is to place 8 cats in 8 cages for 2 hours a day for 5 days; with each cage having two bowls of different food products. The initial and the final weights of the food in each bowl are measured to determine the amount of diet consumed.

The system was recently upgraded to allow real time weighing of each bowl [2]; which introduced several improvements and benefits, such as:

1. identifying the number of meals each cat had;
2. identifying the periods between meals;
3. identifying the amount of food consumed in each meal;
4. identifying the relative rate of consumption of the 2 diets by each cat, which ultimately is a measure of the palatability of each diet.

In addition to these benefits, it was realised that there may be a possibility of extracting new information from the 2 hour testing sessions by monitoring the cats and studying their behaviours, while they consumed their meals. Cats have a complex feeding pattern; they consume small meals at intervals during a day, rather than finish the diet offered to them in one sitting. This type of behaviour makes it impractical for the analyser to directly observe their feeding behaviour because he/she would have to observe the cats for the entire session. In addition, a single person cannot track multiple cats simultaneously. Therefore, to facilitate these types of studies, we created a real time video-based monitoring of cat feeding behaviour system, which has the potential to:

1. Allow the analyser to determine how and where the felines have their meals.
2. Clarify some of the anomalies in the real time weighing system's results, such as the reasons behind the sudden drop or rise in a bowl weight, which cannot be determined without a visual proof; i.e. determine whether the cat may be playing with the food, spilling the

food, or removing the food from the bowls and consuming it somewhere else.

3. Enable the analyser on determining the role of olfaction (smell) in meal choice.

The proposed solution for monitoring cat feeding behaviour was to install a USB camera in front of each cage, which was held in place by a mechanical arm that provided the flexibility in adjusting the angle and the distance of view. Then a PC was used to run Lab View software that performed a background subtraction algorithm to each cage simultaneously at 25-30 frames per second to detect whether the cat was in close contact with the food or not. At the same time the program recorded the real time weight of each bowl and associated each weight reading with its corresponding frame while recording.

A separate Windows Presentation Foundation application was run on the client machine which presented the collected data to the client or the analyser with an elegant user interface; where the analyser has the ability to navigate through each recorded video by clicking on any value in the real time weight chart *(more details are given in sections 3 and 4).*

The challenging aspect of the monitoring system was the implementation of a motion detection algorithm that was computationally inexpensive and at the same time worked for cats of various colours *(apart from white cats, since the cages' background is white)* while keeping the number of false alarms to a minimum. The background subtraction algorithm was modified to account for slight changes in lighting, from cloudy to sunny, while not detecting small changes in background, such as a cat's tail swinging in the feeding area, small vibration of the camera and so on.

2 RELATED WORK

The feeding behaviour monitoring system consisted of an activity triggered video recording system and a customised media player to display the obtained results. During our research, it was found that there were several works and solutions already published using video-based tracking systems.

Chen et al. [3] developed a stand-alone video-based animal tracking system for noiseless application. Their approach for tracking rodents in an acoustic environment was to fix a coloured marker to each target object. Then a tracking algorithm was implemented, that used the Y (luminance), Cb (blue-difference chrominance) and Cr (red-difference chrominance) values of the coloured marker to track the object position and to allow multi-object tracking against a complex background. An FPGA is used to process the colour video signal and to convert the data into sets of coordinates representing the momentary positions of the targets. The information was then stored on an SD card using a microcontroller.

Using an FPGA module, in our project, to perform the detection and the video compression would dramatically reduce the strain on the PC allowing it to handle more cages at once. However, in our case the FPGA module must transfer the compressed video and weights rather than position coordinates, which then must be stored in a hard drive (saving onto a hard drive will limit the number of parallel video streams able to be stored at the same time, thus limiting the number of cages that can be handled using a single PC). Therefore, for the cat feeding behaviour monitoring system, it was realised that using a single PC to handle the detection and the video compression of 4 cameras simultaneously was more economically viable.

Yang et al. [4] tracked multiple workers on constructing sites using video cameras. They addressed the challenge of having multiple workers in an interactive workplace, by developing a tracking algorithm based upon machine learning methods. The algorithm required several sample templates of the tracking target and learnt a general model that could be applied to other targets with similar geometry.

Straw et al. [5] worked on multi-camera real-time 3D tracking of multiple flying animals. Their work was capable of tracking the position and body orientation of flies and birds. They tested their system by using 11 cameras to track 3 flies simultaneously at 60 frames per second in a 3D space using a gigabit network of 9 standard Intel Pentium 4 and Core 2 Duo computers.

The work presented by Yang et al. [4] and Straw et al. [5] have a limited application to our problem, because the degree of complexity employed is not necessary to detect and record the cats inside their cages. Furthermore, one of our objectives was to build an efficient algorithm that allows a single PC with a standard i5 processor to handle up to 4 cameras simultaneously.

3 DETAILED DESIGN

The major objective of the system for monitoring cat feeding behaviour was to record the cats only when they were in close contact with the food and to match the real time weight data with its corresponding frame number while recording. The algorithm had to be fast enough to enable a single i5 processor to handle 4 cameras simultaneously.

The first step as (shown in figure 1) was to recognise the connected USB cameras and to match each camera ID number with its associated cage number as described in section 3.1. Then a background image must be taken from each connected camera, with all of the bowls inside the cages, and all of the cage doors closed before the system was initialised. Then the system was started by the user, the cats were put inside their cages. This ensured that the initial meal was recorded.

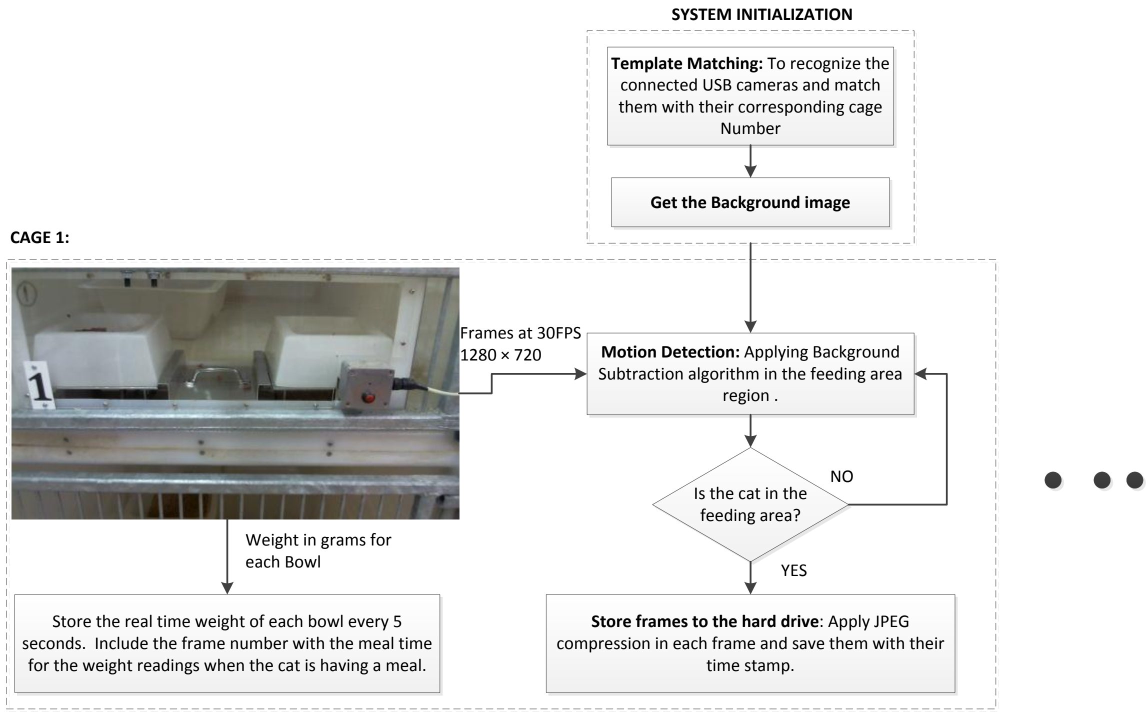


Figure 1: Block diagram of the real time monitoring of cat feeding behaviour system

When the monitoring system was started, a background subtraction algorithm was applied to 4 cages simultaneously and the weights from the 16 bowls (8 cages and each cage had 2 bowls) were recorded.

3.1 *Cage identification*

The cage identification numbers were determined through template matching using the cross-correlation method. The RGB image was first converted to a grayscale image (Luminance, Y) using the following equation:

$$\mathrm{Y} = 0.2126\,\mathrm{R} + 0.7152\,\mathrm{G} + 0.0722\,\mathrm{B} \qquad (1)$$

Then the search region, where the template matching was performed, was reduced (shown in figure 2) to speed up the process. The template images, which were the number plates, were retrieved beforehand and were stored on the hard drive as a part of the system data.

The monitoring system performs template matching 8 times (once for each template image) on each connected camera. The cage number associated with the camera is the number of the template with the highest score. However, if the maximum template matching score is below 50%, this indicates that the selected camera is not connected to any cage.

Figure 2: Search region of template matching

Template matching used correlation which treated the template image as a mask and moved the mask over the image (or the search region) while computing the sum of products at each location. The cross-correlation of a mask $\omega(x, y)$, with an image $f(x, y)$ may be expressed in the form:

$$c(x, y) = \sum_{s}\sum_{t} \omega(x, y)\, f(x + s, y + t) \qquad (2)$$

Equation 1 is very sensitive to scale changes in f and ω; so instead a *normalized correlation coefficient* was used, where values vary from -1.0 to 1.0, as described in Gonzalez and Woods [6]. The location where the coefficient is at its maximum and above 0.5, was where the match occurred.

3.2 *Motion Detection*

Motion detection was performed using a background subtraction approach. Background subtraction was applied to the luminance of both the background image (BG) and the luminance of the new frame (F). Both the BG and F images were cropped to include only the food area which was the region of interest. In addition, the process speed was improved by reducing the resolution of BG and F images by 2 in both horizontal and vertical directions. Then the absolute difference of F and BG was calculated:

$$\mathrm{Diff} = |\mathrm{F} - \mathrm{BG}| \qquad (3)$$

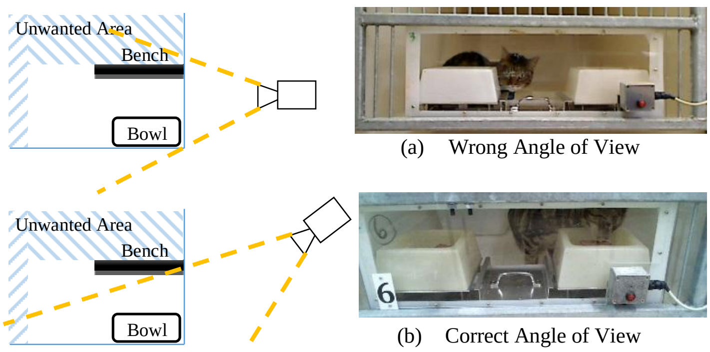
The absolute difference in F and BG intensities was an excellent method to separate the cat body from the background, because in our case there was a sufficient difference between the background (white) and the foreground intensities *(since there are no white cats in the CFN).* This difference was very sensitive to noise and slight changes in BG or F intensities, so most of the small changes in intensity *(camera arm vibrations, small changes in lightings, shadows and reflections)* were avoided by thresholding the absolute difference.

However, a fixed threshold value was not applicable for our application; therefore, it was decided to use optimum global thresholding using Otsu's method to account for cats of different colours, such as black, brown, bright red and orange etc. Otsu's method is optimum in the sense that it maximizes the between-class variance [6]. In other words, when F has a bright red cat or orange cat, the total mean value of the absolute difference intensities will be low, so the Otsu's method will result on a low threshold value to separate the cat from the background noise, and will be the opposite for the cats with dark colour. Therefore, using an optimum global thresholding approach is like having an adaptive threshold that adapts for cats of different breeds and hair colour.

The next step was to decide whether the cat was in close contact with the food or not. This presented several challenges: 1. Cats often sat on the top of the feeding station and their tail occasionally dropped into the feeding area region which had the potential to cause false frame acceptance. 2. Cats occasionally sat or slept behind the feeding area in the cage, which was in the camera field of view and therefore within the region of interest.

The first issue was addressed by applying morphological filters to the thresholded image. A morphological closing of 3×3 structure element (SE) with 2 iterations was applied, followed by a morphological opening of the same SE with 5 iterations. This combination of morphological filters minimised the area of the tail while preserving the majority of the cat's body.

The second problem was solved mechanically by titling the camera's angle of view (shown in figure 3). This prevented the camera from capturing the cat at distances greater than 40 cm away from the food, and at the same time, it enabled the analyser to observe the cat's tongue and mouth while it is eating (shown in figure 3b).



(a) Wrong Angle of View

(b) Correct Angle of View

Figure 3: Modifying the cameras' angle of view to capture the cat when it is in close contact with the food

The decision of whether the cat was in close contact with the food or not was based on measuring the area (the number of detected pixels) of every connected component within the binary image. If the maximum area was greater than a certain threshold value, the cat was recognised as being in the feeding area otherwise it was not. The threshold value used in our case was 2000 pixels.

The background subtraction algorithm was enhanced by dynamically updating the BG when the cat is not in the feeding area:

$$\mathrm{BG_i} = \alpha \mathrm{BG_{i-1}} + (1-\alpha)\mathrm{F_{i-1}} \tag{4}$$

This had the benefit of updating the BG when there was a gradual change in lighting or when a new small object was added to the background, thus minimising the possibility of having a false acceptance or a false rejection.

3.3 Camera Arm

The camera arm (shown in figure 4) provided flexibility in establishing the correct distance and angle of view, thus making it suitable for different types of camera. In addition, the camera arm was long enough so the camera was out of reach of cat's claws.

This design had the utility for permanently fixing the rotation of every joint; the first joint had 3 holes where the arm could be bolted, while the angles of the second and third joints were fixed using a bracket (shown in figure 4). This ensured that the ROI and the angle of view (as discussed in section 3.2) remained consistent.

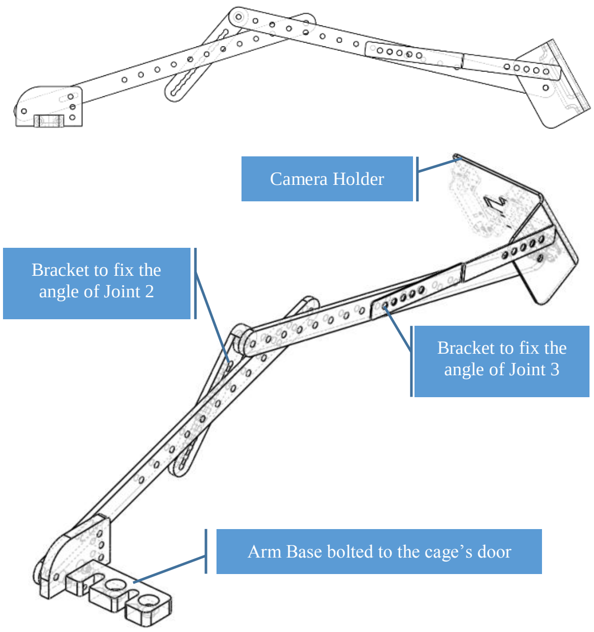


Figure 4: 2D and isometric drawing of a Camera Arm

3.4 Real-Time Weighing of the food bowls

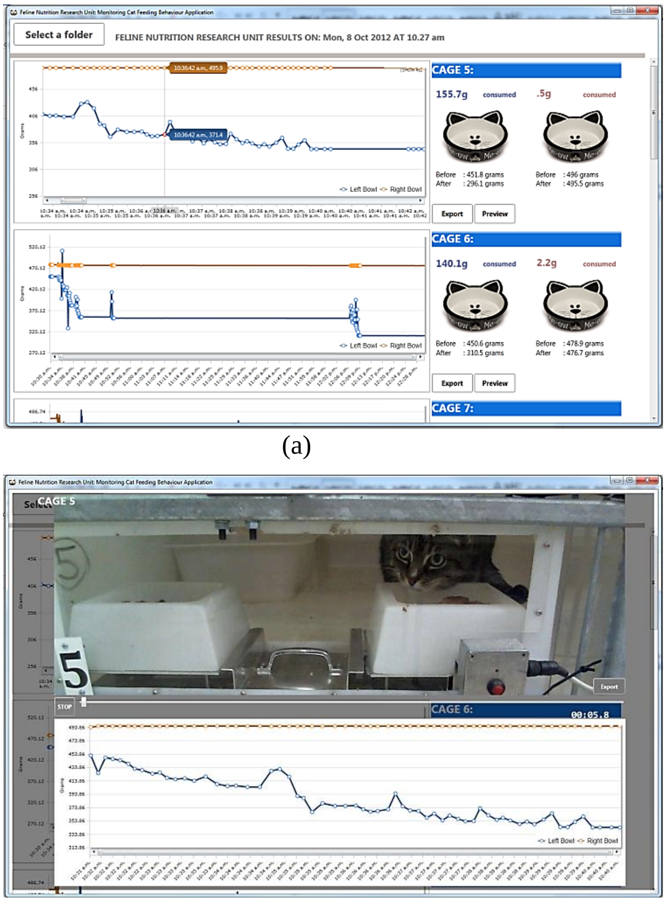
The weight of each bowl was measured using a load cell, where it's output signal was first amplified using *Instrumentation Amplifier INA2126 IC*. Then the amplified signal was fed to a 12 bit analogue to digital converter. The digital signal was then passed to a microcontroller, where the input voltage was related to its corresponding weight in grams. After conducting several experiments on the load cells, it was found that there was a linear relationship between the amplified voltage and the real life weight (grams), as shown in the following equation [2]:

$$\text{Weight (g)} = 502.40 \times \text{Voltage (V)} - 570.38 \qquad (5)$$

A test was conducted to estimate the errors of the load cells, when equation 5 was used. The errors when plotted in x-y scatter plot appeared to be random and did not follow any particular pattern. In addition, the largest error was 0.782g, which was within the acceptable range of this process.

The microcontroller communicated with the host PC using the USB port. The PC accessed the weight of every cell, by transmitting the cage number to the microcontroller, and waiting for the microcontroller to process and send back the real time weight. This approach allowed the PC to access the weight of each load cell independently.

3.5 Presentation of the collected data to the client



(a)

(b)

Figure 5: Screenshot of the client software

The data collected from the experiment were the recorded videos and the bowl weights. This information was presented to the client with an elegant user interface application (shown in figure 5) with the following features:

1. The main window provides a quick overview of the result of each cage. A taskbar shows a summary of the amount of food consumed from each bowl and a complete 2 hour chart of the weight.

2. The user can click on any circular point on the chart to display the recorded video of that cat at that time.

3. The video player was designed to display the frames at their corresponding frame rate using the time stamp. In addition, it has a unique feature where the user can view the cat's behaviour associated with the bowl's weight by clicking on the corresponding point on the chart (shown in figure 5b).

The software was designed using Windows Presentation Foundation tools. The user first needs to select the folder that he/she wants to analyse (folder names contain the date and the time of the results). Then the software will read the weight charts for every cage and attach the recording time and the frame number for each point that is within the recorded period. These points are identified to the user by having big round circles (shown in figure 5a).

Clicking on one of these points will open up another window (shown in figure 5b). This window is the media player, where the video of the selected point will be played; the user has the option to stop, pause, replay and use the cursor to navigate through the video stream. Moreover, the system will display a detailed weight chart underneath the taskbar. Each point in the chart has the frame number attached to it, so it is possible to navigate through the video by clicking on any point in the chart.

4 EXPERIMENTAL RESULTS

4.1 Aim

The aim of the experiment was to assess the performance of the system and to understand how it will help the analyser assess visual components related to the cats' preference of the tested diets.

4.2 Method

The system was installed at the Centre for Feline Nutrition and it was used to monitor 4 cats simultaneously for 4 hours a day (2 sessions, with each session 2 hours long), for 5 days. Then the results were analysed by the authors on a daily basis.

4.3 Results of System's Performance

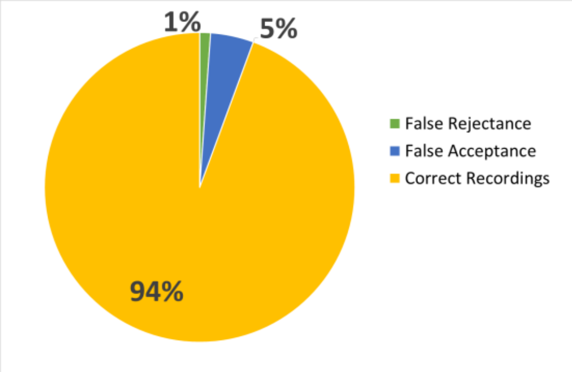


Figure 6: Pie Chart that shows how successful the system was in detecting the cats while they consumed their meals.

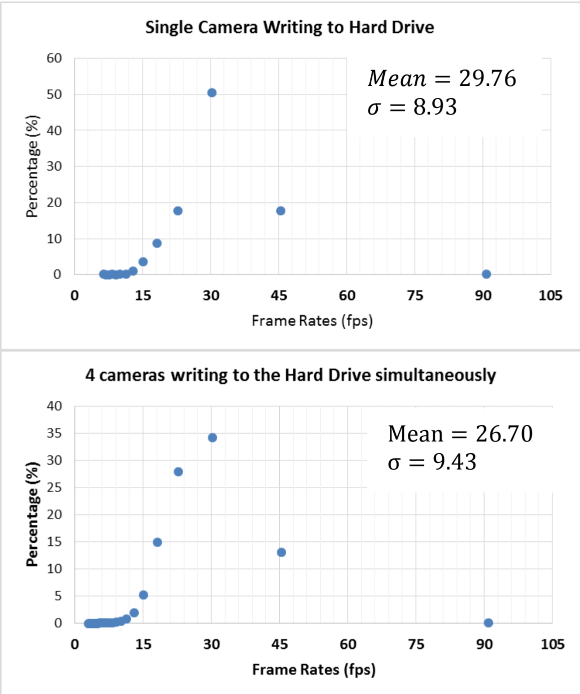


Figure 7: Frame rates versus the percentage of the total frames that were recorded. The system was tested in Intel i5 Processor with 8 GB RAM and 7200 RPM Hard Drive.

4.4 Discussion

4.4.1 System performance

The performance of the motion detection algorithm was evaluated (shown in figure 6). False rejection (FR) was identified by recording a single frame every second and using these frames to search for any movement that the system failed to detect. It was found that the system had 1% FR. This occurred when a cat had a meal of less than 4 seconds. This movement was recorded, but the system deleted it afterwards, since it was programmed to discard all recordings that were less than 4 seconds duration. The majority of the false acceptance was when a cat jumped from the bench to the back of the cage more than once during a 10 second period (the system was modified to have 10 second buffer after the end of each detection).

The average recording frame rate dropped from 29.76 to 26.70 (shown in figure 7), when the 4 cameras were recording simultaneously. This was due to the limitation on how fast the system could write to the hard drive and on how much CPU resources each JPEG compression required. The size of each JPEG image could be reduced by building a customised compression that exploited the temporal redundancy between frames. However, a customised video compression would increase the computational complexity of the system and a single PC might not be able to accommodate 4 cameras at once.

4.4.2 Analyser's experience with the new system

During these 5 days the analyser was able to watch the cats while they consumed their meals and to observe their behaviours. Moreover, there were several occasions when the cats rejected certain foods because of their smell, information the Centre had been unable to access previously.

4.5 Conclusion

The system performed well with a successful detection rate of 94%. It dropped an average of 3 fps when 4 cameras were recording simultaneously, which was an acceptable frame loss. In conclusion, the system served its purpose by enabling the centre to monitor the cats.

5 CONCLUSION

Real time monitoring of cat feeding behaviour system enabled the CFN to access new information from the 2 hours testing session that was unattainable before. The system will assist the Centre in understanding some of the reasons behind the cat's rejections and acceptance of certain food products, thus providing more feedback to the pet food manufacturer in relation to their products.

6 REFERENCES

[1] New Zealand Pet Food Manufacturers Association (2012). http://www.petfoodnz.co.nz/ (retrieved on 27 April, 2012)

[2] M. Cho. (2009). *Automated Analysis of Feline Feeding Behaviour*. Engineering Honours Project Report. Palmerston North: Massey University.

[3] Y.-J. Chen, Y.-C. Li, K.-N. Huang, S.-L. Jen, and M.-S. Young. (2009). Stand-Alone Video-Based Animal Tracking System for Noiseless Application. *Instrumentation Science and Technology*, vol. 37, no. 3, pp. 366-378.

[4] J. Yang, O. Arif, P.A. Vela, J. Teizer, Z. Shi. (2010). Tracking Multiple Workers on Construction Sites Using Video Cameras. *Adv. Eng. Info*. vol. 24, no. 4, pp. 428-434.

[5] A.D. Straw, K. Branson, T.R. Neumann and M.H. Dickinson. (2011). Multi-camera Realtime 3D Tracking of Multiple Flying Animals. *Royal Society Interface*, vol. 8, no. 56, pp. 395-409.

[6] R.C. Gonzalez, and R.E. Woods. (2007). *Digital Image Processing*. (3rd, Ed.) Prentice Hall.