**ML applications in fall detection for the elderly**

**Aengus Leman1, Eldora Boo Puay Eng2, David Tan Yue Hua3, Wong Jia Xi4, Qin Guorui5**

National University of Singapore1,2,3,4,5, CS3244 Group 41

A0216100E1, A0221833M2, A0216417J3, A0190110B4, A0219697J5

**Abstract**

Falls remain to be a serious threat to the well-being and lifespan of our elderly in Singapore. Fall detection has been a way to mitigate negative consequences from falling, as the duration between the actual fall and medical attention can be the difference between whether the elderly lives or dies after a fall. We propose a mobile app for caretakers that uses the faster R-CNN model to detect if high risk individuals have fallen down from a video feed. Faster R-CNN is a regional convolution neural network that is widely used for object detection tasks.

To test the efficacy of our model in detecting falls from a given video feed, we labelled a dataset containing 24 different scenarios of individuals falling, including potential edge cases. After labelling, we used the dataset to train our model. The model produced had an average precision of 90.2 at an IOU of 50%, which we determined was sufficient to use in the app.

After 10000 iterations of training the model, our model achieved an average precision value of 90.2 when tested on the test dataset. As this is relatively high, we deem the usage of Faster RCNN a success in fall detection.

**Introduction**

The statistics are shocking to say the least. It has been reported by the Singapore Medical Journal that “approximately 28-35% of people aged 65 and over fall each year increasing to 32-42% for those over 70 years of age.” 1(Medrano 2013). Many of these cases are in fact labelled as “silent fallers”, where their falls go completely unnoticed and the elderly have to deal with it on their own. A whopping 40% of injury-related deaths in Singapore are due to falls. As such, we can see why this is a pertinent problem in Singapore.

Fall detection in Singapore has always been seen as a way to mitigate some of the negative consequences from falling. More specifically, reducing the time the elderly spend on the ground, also known as “long lie”. Seeing the falling possibly inhibits one’s ability to get up from the ground, the dangers of falling are also extended by how long the person spends on the ground. Possible complications include hypothermia, dehydration or even pressure sores. The bottom line is that if left untreated after a fall, the health risks to an elderly person could increase tenfold. Hence, the research we are conducting is more pertinent than ever in Singapore.

The main target audience are caretakers of the elderly, more specifically those caring for individuals living in old folks homes or living alone at home. We chose to target this group because elderly consist of the bulk of the victims of falls in Singapore. Furthermore, they are more likely to be unable to get up or help themselves after a fall hence it would be more prudent to target them. The target audience in question is important because we need to train our model on specifically elderly figures in certain locations, although we have the datasets including people falling with a mix of ages and locations. This evades a bulk of false positives, for example if a child were to fall down often in the area. This is reflected in the dataset we have chosen, which would only include elderly falling down being marked as positive. We have also narrowed down the usage of our system to mainly indoor, private spaces such as living rooms, kitchens, outside toilet areas etc. Limiting our dataset to indoor spaces reduces the likelihood of incurring erroneous detections so any possible complications from data outdoors would be minimised.

Our proposition comes in the form of an app catered to such caretakers which implements a machine learning model to detect falls of the elderly. The implementation of our model uses constant surveillance of set areas, where the outlines of people walking within the area would be highlighted. This is done via feature extraction, using the ResNet 101 Convolutional Neural Network (CNN). Following that, if an image of an elderly falling was detected, the model would detect that and send an output to alert relevant (potentially multiple) caretakers or authorities, also optionally providing a video feed to the caretakers when there is a fall detected.

**The Current Situation**

Currently, one of the biggest problems is implementability, due to the limitations of the datasets. Many of the datasets were centred around elderlies carrying out simulated activities, due to a lack of organic data. Either that, or the data consisted largely of young people rather than older people. As such this took a toll on the eventual accuracy of the trained models. We instead opted to look for data of older people falling down online, and labelled the data ourselves.

One particular report done by Liu,2 showcased the use of K-Nearest Neighbors. They chose to divide the motion of the elderlies into Backward, forward and sideways falls. We found that doing so turned out to be ineffectual, and did not contribute to the accuracy of the model, hence we took note not to discriminate between the types of falling in order to improve accuracy.

Furthermore, most models rely on expensive technology such as multiple 3D cameras, motion sensors, pressure sensors and infrared cameras (in addition to existing processing units) to obtain a high degree of accuracy in monitoring. Our model simply relies on vision (requiring a simple camera) and a processing unit to detect a fall which provides a much cheaper alternative for those who cannot afford such technologies but still would like to protect their loved ones.

Following up from this, we will elaborate on how the models and training methods we have chosen actually help to solve these issues.

**Understanding the Model**

Our team proposes to use the faster R-CNN model to detect images of people falling down in the CCTV that will be placed in homes of people who are at high risk of injuring themselves.

Faster R-CNN is a regional convolution neural network that is used widely for object detection tasks. In its architecture, there are many features that will aid us in improving our results quickly and efficiently

**Introduction to Faster R-CNN**

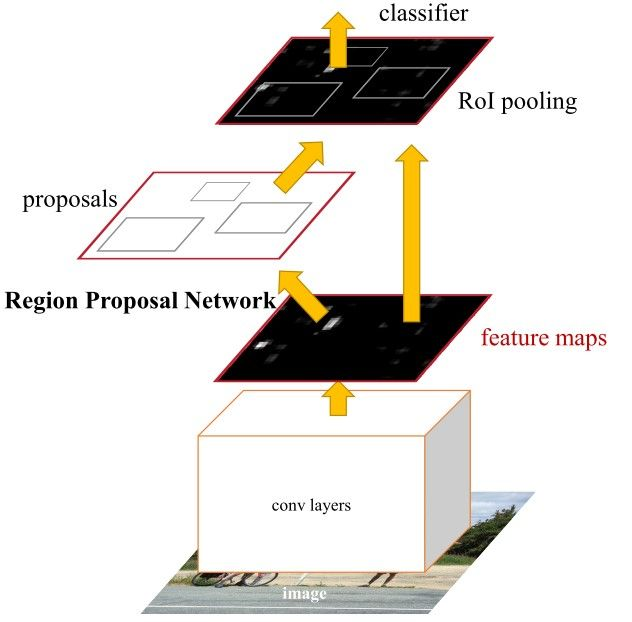


Figure 1: Faster RCNN network (Gad, A. F)

To understand how image detection works, reference was taken from authors of Faster R-CNN (Ren et al. 2017 . Within the Faster R-CNN framework, it is mainly made up of three portions, a feature network (usually a CNN to extract out feature maps), a RPN to detect object bounds and object scores, a Region of Interest (RoI) pooling layer, then followed up by a series of fully connected layers used for classification.

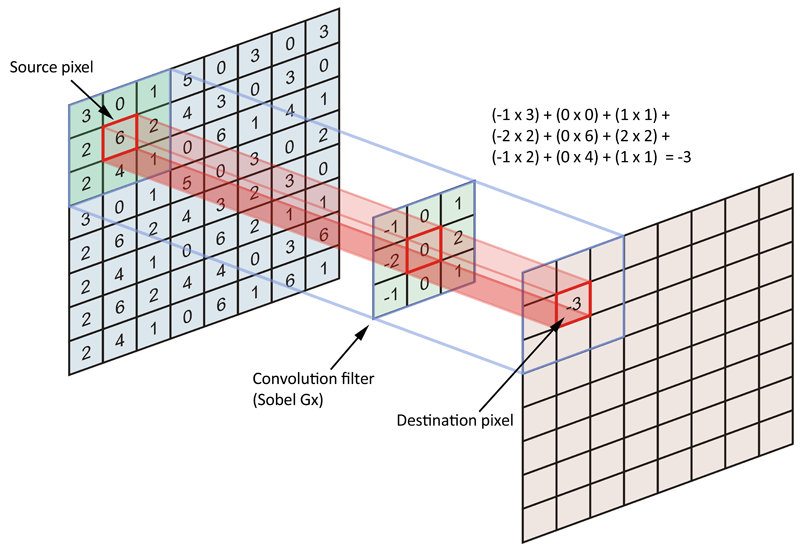


Figure 2: Convolutional Layers (Data Science Stack Exchange, 2017)

In the feature extractor network, convolutional layers are used. Figure 2 shows an example of how convolutional filters work. Each grid in the image is then convolved with a network trained filter to produce an output in the feature space.

The feature extractor usually consists of a pretrained network such as ResNet which was previously trained on another classification task. By using pretrained models, it allows us to use weights that are close to our target rather than starting from randomly initialised weights. The goal of this network is to generate important features to be fed into the RPN.

The feature extractor uses convolutional layers in its network which is better than simple forward feeding networks. In a simple forward feeding neural network, an image of different pixels is usually resized to the target size, then followed by flattening them into 1d vector to save computation times. However, in convolutional layers, the image is fed through a filter or a kernel, so that faster computation time is achieved without the loss of features.

After the image is fed to the feature extractor, the output feature map of the image can be used for our RPN. In the RPN, will then give a prediction on where the object is located. In turn, the RoI pooling layer will extract fixed-length feature vectors for all proposals generated by RPN in a single feature map. Finally, the extracted proposal regions will be feeded into fully connected neural network layers to predict the class of the object.

**Comparing convolutional neural networks to a simple forward feeding neural network**

Using only convolutional neural networks as compared to a forward feeding neural network brings about many advantages.

Firstly, performance of convolutional neural networks is superior to that of a simple forward feeding neural network. The convolutional layers provide a deeper analysis of the image. In the initial layers of the networks, colours and edges are detected while in the later layers, it can be trained to identify certain patterns in the network.

Secondly, with max pooling and convolutional layers,the number of inputs will be reduced each layer, with better higher level information within all the inputs. The number of trainable parameters would be less for a neural network with the same number of hidden layers. This will greatly reduce the time taken to train and optimise our model.

**Interaction with application**

Our application is used to detect a specific situation of falling down of the elderly, and it’s installed in a surveillance camera of indoor spaces, such as kitchen, living room and so on. This application can detect image by image in real time. Users, such as nursing staff, will be informed (either through bell ringing or light shining) if the falling situation occurs, and they can check with the specific frame detected as falling down to ensure the elderly are really falling down (maybe there’s a false positive detected). As such, they can make decisions like whether to leave it if the elderly just lie on the bed to have a rest or take the instant reaction to a serious falling down. As our detection method seldom has false negatives (actually falling down but not recognized), it could still be a good choice to ensure the safety of the elderly to even double check the possible false positives.

**Justifying the Choice of Models**

We decided to use Faster R-CNN in our project as it presents the fastest results out of all the object detection algorithms. We will be comparing it with its predecessors, R-CNN and Fast R-CNN.

Both R-CNN & Fast R-CNN use the selective search algorithm to find out the region proposals, which is a slow and time-consuming process that affects the performance of the network. The selective search algorithm is a fixed algorithm and no learning is happening at that stage, this could lead to the generation of bad candidate region proposals.

So in Faster R-CNN, an object detection algorithm that eliminates the selective search algorithm is implemented and lets the network learn the region proposals.

In Faster R-CNN, a feature extractor network is used to predict the region proposals instead of the selective search algorithms. The predicted region proposals are then reshaped using a RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes. Because Faster R-CNN is so fast, it can be used for real-time object detection.

For the feature extractor network, we are using ResNet for this segment. We will compare it to its counterparts, AlexNet, VGG, and GoogleNet, which have 5, 19, and 22 convolutional layers respectively. However, deep networks are hard to train and cannot be done by simply stacking more layers together. As the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient smaller and smaller. Hence, as the network goes deeper, it results in saturated and degraded performance.

ResNet introduces an “identity shortcut connection” which protects the network from the vanishing gradient problem. The gradients can flow directly through the skip connections backwards from later layers to initial filters. It also utilises Batch Normalisation at its core, which adjusts the input layer to increase the performance of the network. Lastly, the bottleneck residual block design increases the performance of the network.

The models we chose fit the requirements of object detection technology in our proposed application. Our proposed application seeks to detect people falling and it is trained to be able to do so. They have worked well in detecting falls but it ends up capturing false positives as well.

Although we accounted for objects surrounding the subject in order to determine whether it is an actual fall, or actually a person lying down, there could be other edge cases that we fail to account for. However, it was mostly successful in detecting falls in general on an open floor area.

**Ideas for improvement**

We are aware that within the scope of our project, our current model is relatively simple compared to many state of the art technologies out there. There were many significant ideas mentioned within our team to potentially improve the performance of our model.

One weakness of our model is its propensity to have a high false positive rate. Our model primarily detects if an elderly has fallen by extracting features from an image. However, if an elderly chooses to lie down or fall asleep on a bed or couch in a pose that is similar to a fall the model would likely send a false positive. We ideated using this model in conjunction with other ensemble methods such as motion detectors and accelerometers. These methods generally boast a higher predictive accuracy . However these methods also come with additional monetary costs, as well as potential intrusiveness (in the elderly constantly having to wear motion devices etc).

Another suggestion is to train the model to ignore surfaces (not including sides) meant for lying down: stationary beds and long sofas etc to prevent false positives. In the case that the elderly actually falls on such a surface, it is a lot less likely that there would be a serious injury anyway (under the assumption that most of these surfaces are soft and therefore less dangerous).

If the user is able to install several cameras to provide different angles of an area, there could be an additional hidden layer in the neural network to consider the multiple images being fed into the model at once. Alternatively, the model could simply produce an output based on a majority vote of the different angles. Since installing multiple cameras in one location is not a common assumption, we primarily built our model to function with one visual source.

During the surveillance, the application installed in camera itself can self update its dataset when it captures more useful frames like a real time falling down of the elderly, this means the application could become more experienced if it’s used for longer time, this may be achieved when it detects a real time falling, but the nursing stuff thought it’s a false positive, then it can label “not falling down” to this “falling” frame it detected, and the knowledge base of the application has been improved, and once, for instance, 20 such situations accumulated, it will self update the dataset and re-train the model in a short time. As long as the new model is trained it can replace the original one to improve the classification accuracy and lower possibility of false positives.

**Choosing the dataset**

Our proposed application should rely on a relatively large set of data to ensure less false positives and negatives. The images could be quite complicated if it involves different furniture and colourful backgrounds. Even after using RPN to figure out the proper region for classification (human body in this case), different poses, ages, types of build or even expressions and genders of people could complicate the classification process. Under a small dataset, it’s very likely to result in an overfitting. For instance, if we only have the elderly wearing red clothes falling down on the red floor labelled as positives and the elderly wearing blue sweeping the blue floor labelled as negatives, the backbone resnet may learn the major internal distribution of these images, and recognize “red” images as “falling down” and “blue” images as “ok”. The issue occurs when an instance of the elderly wearing red sweeps the floor occurs, the resnet will classify it as “falling down”, and a false positive occurs. As such, it really requires sufficient data for each class of “sit down”, “fall down”, “lie down” and so on in order for our model to make the correct judgement, and our model can deal with the real time frames and big data due to its efficient way to extract out the key features of each object region and the relatively short computing time to classify the detected object.

We decided to use this particular video set (E. Auvinet et. al. 2010) for the following reasons: The dataset was very detailed, providing 24 different video scenarios (22 of falls and confounders, and 2 of strictly confounders) for training (resulting in 2248 images). This allowed us to train on a wide variety of cases such as if the subject were to lie down to rest (not a fall, see Figure 3), if the subject were to fall down on an object such as a table (falls in strange angles), if the subject were to bend/kneel forward to pick up a dropped object (not a fall) etc. Training based on these very possible potential confounders with a decently large variety of scenarios would increase the accuracy of our model in these actual situations. A few flaws of the dataset are as follows: The entire dataset was male dominated, there were not many instances of females falling. Only one room was used, however the furniture was moved around to simulate falls on/beside different furniture (such as falls on table/falls off of chairs)

The entire dataset of 192 videos had to be extracted into images (one image/frame per second), and then hand-labelled, labelling the x-y coordinates of the objects (people in the frame), with a negative label being not fallen and a positive label being fallen. 

Figure 3: Example where resting on couch that may result in a false positive detection (fallen instead of non-fallen)



Figure 4: Example of a normal positive training instance (falling/fallen)

**Experimental Setup**

All our experiments were done on google colab using their GPU (NVIDIA Tesla K80). Frames that do not include any humans were removed from the dataset, resulting in a total of 1759 images with labels. As training the model requires a significant amount of time, we simply split the dataset into train, test and validation to select the best model instead of doing k-folds cross validation or leave one out cross validation.

Amongst the 1759 images, 1231 images were used for training, 352 images were used for testing and 176 images were used for validation.

Model weights were saved every 1000 iterations. Training of the dataset was carried out for 10000 iterations and the model with the highest mean average precision (mAP) score tested on the validation dataset was used for our application.

**Ethical Implications of the Study**

The concept of surveillance is one that comes with a host of ethical grey areas. Would the privacy of those being monitored be breached as a result? Would it be ethically acceptable to carry out said monitoring for the sake of safety? One of the places where falls tend to occur the most is restrooms, and it would definitely pose an issue if cameras were to be placed in the restrooms.

Another ethical issue that stands out is whether surveillance truly benefits the elderly. One key aspect of having such a system is to provide a sense of security for those under it. However, some might receive such a system with the opposite effect, feeling that they are constantly being monitored. Also, some might refuse to let the camera record some images for further upgrading and learning purposes since these could easily be regarded as an invasion of privacy.

**Results**

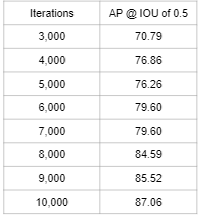
****

Figure 5: Table of results of model

After running each model through our validation set, we found that the model at iteration 10,000 achieved the best AP result of 87.06 when tested on the validation set.

This model also gives an AP of 90.2 when using an IOU threshold at 0.5 on the test dataset.

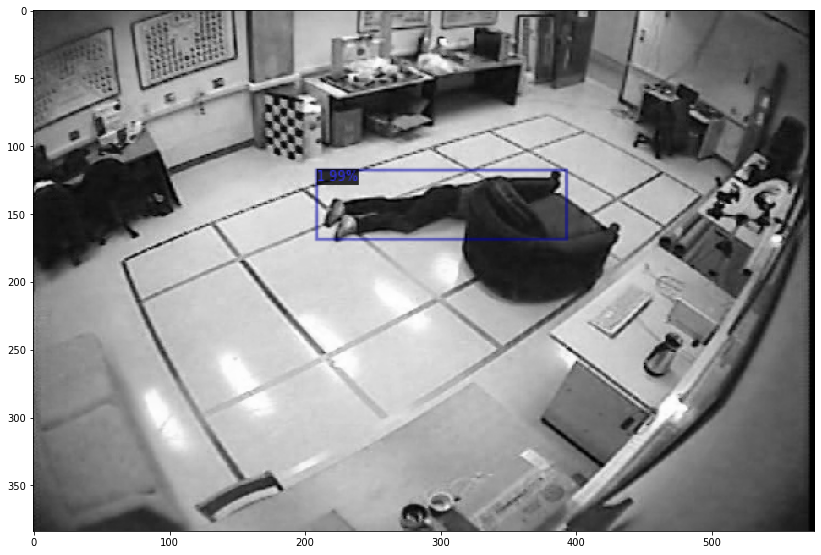
****

Figure 6: Example of a fall being detected

****

Figure 7: Example of no falls being detected with humans in a room

**Concluding Thoughts**

We had the more experienced machine learning members spearhead the writing of the more technical aspects of the project, such as creating and training the model, while the other members researched the current state of fall detection and contributed to other portions of the report. All members got to try labelling and training the model on our own computers.

Aengus: From this project, I have gained a greater understanding of the process that goes behind training a machine learning model, and how the code should go. Learning about different types of machine learning models, such as the Faster RCNN with the detectron 2 library was also interesting and understanding how it is relevant for usage in computer vision was enlightening. Finally, the data was also extremely difficult to label, and my group has gained a new appreciation for labelled data.

David: This type of data is excruciatingly hard to transform into supervised data, i.e. label by hand. It requires object detection as well as labelling of the output label for each training example. We experienced firsthand why (labelled) data especially in large quantities can be considered fairly expensive, and have learnt the importance of appreciating limited data by learning how to use it well.

Eldora: I learnt that it is important to realise that there will be gaps between what the model is able to predict for us and what ends up happening in real life. We will always have to take into consideration the possible confounders during the testing, and adjust our model accordingly for an optimised result.

Guorui: From this project, I learned the basic structure and application of Faster R-CNN concretely, understanding how it could be improved and designed based on previous work of Fast R-CNN. Also, I understood how to import this model structure to perform machine learning training on our own datasets and apply the organisation of the validation set and testing set learned during the lecture.

Jia Xi: From this project, I have learnt the way to implement Faster R-CNN with the aid of the detectron 2 library which was fairly intuitive to use. With the help of my team, I was also able to use the data extracted and manipulate their labels into the desired format (COCO).

**Roles**

Aengus: Narrowed down the idea of fall detection, wrote the introduction and current situation, ideated improvements to the model, helped to train the model and discovered several possible ethical implications.

David: Collated ideas for improvement, found suitable dataset, organised labelling of data, research and writing for current situation of fall detection.

Jia Xi: Wrote scripts to convert data formats and did a notebook to train the object detection model. Covered the architecture and experimental setup components of the report.

Eldora: Wrote the current situation, justifying the choice of models, making technical insights

Guorui: Analysed the dataset requirement, described how actual interaction with the application happened and checked the overall writing and code implementation

**References**

Medrano, C., & Plaza, I. (2013, July 6). Challenges, issues and trends in fall detection systems - BioMedical Engineering OnLine. BioMedical Engineering OnLine. Retrieved April 15, 2022, from <https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-12-66#ref-CR1>

Liu, C.-L., Lee, C.-H., & Lin, P.-M. (2010, October 14). A fall detection system using k-nearest neighbor classifier. ScienceDirect. Retrieved April 15, 2022, from <https://www.sciencedirect.com/science/article/abs/pii/S095741741000285X>

(n.d.). Multiple cameras fall dataset. Retrieved April 15, 2022, from <http://www.iro.umontreal.ca/~labimage/Dataset/>

Ganyo, M., Dunn, M., & Hope, T. (2011, February 7). Ethical issues in the use of fall detectors. Cambridge Core. Retrieved April 15, 2022, from <https://www-cambridge-org.libproxy1.nus.edu.sg/core/journals/ageing-and-society/article/ethical-issues-in-the-use-of-fall-detectors/ED7DAA8268F914C9EC4C7D412A6EFCE3https://arxiv.org/abs/1506.01497>

Gad, A. F. (n.d.). Faster R-CNN Explained for Object Detection Tasks. Paperspace Blog. Retrieved April 15, 2022, from <https://blog.paperspace.com/faster-r-cnn-explained-object-detection/>

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/tpami.2016.2577031

Why convolutions always use odd-numbers as filter size. (2017, September 20). Data Science Stack Exchange. Retrieved April 15, 2022, from <https://datascience.stackexchange.com/questions/23183/why-convolutions-always-use-odd-numbers-as-filter-size>