Logbook 4th Year

End of Third Year

- Using TLD to remove many steps from last year.
- First priority is to research and understand this, Toby has sent his lecture slides for this.
- TLD should work with any camera and any angle
- Should work with any targets (human and non human)
- Look up if anyone has used TLD for Re-ID in colour or thermal.

Summer Notes

- Want a bash script to cmake, make and run for different purposes (testing, single camera, multi camera, datasets etc) to improve usability from late year.
- Maybe have a command line parameter for starting frame and FPS for testing in specific situations
- Can I just reuse the related work section from last year and add the TLD paper?

Start of Code

- Have used old MoG and Cascade code, along with the OpenCV TLD implementation for a first pass.
- Currently only has one tracker, have read about Multitracker but not implemented yet. Have set up variables
- Currently very slow.
- Experimented with removing backrground from bounding box, this doesn't work with TLD as it then interprets any foreground pixels as the target.
- Currently relatively happy with how it functions on one target, need to find a way to get the system to add more targets.
- Tracking component of TLD is the Median Flow algorithm

- Need to find some way of getting confidence level, seems to be a Featureset attribute but currently unable to access this.
- MultiTracker now works, but the targets are still too similar.
- update() and update_opt() both are not discriminative enough, but update_opt is slower.
- Downloaded a couple of alternative TLD implementations from git.
- MultiObject TLD implementation from git seems good on initial webcam tests, have to try implementing with my thermal videos to see if it is up to the task. Also need to understand what all of the different boxes around a match are, as some appear on incorrect matches but not all. Note - the classifier is not loaded from the file name "test.moctld", they have not yet implemented this functionality.
- Tested MultiObjectTLD implementation with myself as the first target, and only that target. It is still misidentifying Toby and Greg as me, like the OpenCV implementation, but seemed to have fewer of the bounding boxes around them.

Meeting 1

- Research Machine Learning for Re-ID, use this to write related work section of paper (lit review)
- Ask TLD company about how much a development license would be and if it would be suitable (can we extract recognised target)
- New plan is to apply machine learning to the problem to a greater extent than previously.
- Previsionally this will involve an svm per classification of target
  - The TLD implementations that I have experimented with is good at tracking a target through a camera view, but does not perform re-identification well.
  - This implementation does have a NearestNeighbour classifier to attempt re-id, should probably rewrite this component using a better machine learning algorithm, or even just experiment with parameters.

Week 1

- Maybe should investigate openTLD, downloaded and had a play.
- Tried CMT as well, seems to be a good tracker.
- Experimented with other OpenCV tracking algorithms, particularly with KCF, which seemed capable of decent Re-Identification, ask Toby if evaluating multiple algorithms would be a good deliverable.
- Different Pseudocode idea

  - For all targets in image, if any of the svm's have a high enough confidence rate then do not create a new target, and update the TLDMultiTracker, which should get the right one.
  - If the confidence rates are insufficient, make a new target, add new TLD, then update.
  - When displaying the bounding boxes, train the appropriate SVM for the target.
  - Concerns:
    - Different targets for the front and back of people, maybe sides as well

- Got demo from the TLD Vision people, linux version doesn't work at all (doesn't recognise keypresses), windows version does work but the tracking is not as good as the open source implementations.

Meeting 2

- Decided not to do comparison of algorithms as they already exist, have relevent webpage open in a tab.
- Now need to read the relevent papers, recommended by Mendeley, Toby and own googling.
- Need to establish feasibility of online tracking for different of ML algorithms.
- Shortlist of algorithms/TLD implementations to use as base
- Estalish what is going to be extracted from the TLD (or other) to be used to train the ML section.

Week 2

- Check and see if there is existing dataset of thermal profiles that could be used for training purposes.
- Most papers seem to use supervised learning in the training phase.
- Got lots of papers in Mendeley to make my way through now :)
- Can't test struck in its current form as it runs opencv2
- Most of the ML papers have complex CNN, need to understand how I could write this better.
- Need to establish how important runtime is to us, and whether any frames should be skipped.
- Possible Algorithms
  - MOTLD
  - Struck - need to test with OpenCV 2 - currently hasn't been run
  - OpenTLD - look into how to use this - seems to work well(ish)
  - OpenCV implementations - function but slow

Meeting 3

- New idea -
  - Train a CNN on closed world dataset - this should establish good feature vectors for telling people apart at a certain layer
  - Each time we have a new person pass it into the CNN and extract the feature vector at a layer determined by experimentation, and use this feature vector for re-id
  - Compare feature vector using SVM/Mahalanobis etc.
- Papers currently open in browser looking at similar ways of doing this
- could have ML bit in python and then load it into C++
- Next - 3 deliverables - read selection of CNN papers (embedded learning), decide on suitable tracking algorithm, decide on suitable CNN framework - tool - Caffe/tensorflow
- send Toby my ssh public key so I can access the cameras.

Week 3

- This means we are now going to have 2 major discrete components, the tracking component and Re-Id component.
- The Re-ID component will only operate when the Haar cascade has found a target that is not currently being tracked
- Need to ensure that whatever tracking algorithm we choose can handle being told which of their targets to be updated
- Test struck by installing OpenCV2 on Kali box??
- Thoughts:
  - The tracking implementations I am considering make some attempt at Re-ID, need to ensure that they do not try and overwrite the CNN component, but wait to be told which target a new classification is - is motld the best choice for this?
  - Thinking the best solution for testing would be to implement the tracking stage first, and have a key press to replace the neural network for now
  - Maybe pass each target through the neural network every 10 classifications to ensure we have sufficient data for the Re-ID part (SVM or Mahalanobis)
  - Do I want to extract the silhouette or the whole bounding box?
  - Good idea to make all training images the same size or not?
- Keras seems to be a good option, based on either Theano or TensorFlow
- In terms of the tracking algorithm, need to be able to handle multiple objects, but not fight with the Re-ID subsystem.

  - CppMT is very clean and fast, but makes a mistake when greg and I cross over in the exampel video, so probably should not be used.
  - OpenTLD is slower than CppMT, has decent accuracy, would need to use multiple instances of this tracker for multiple objects, also is a very complicated library.
  - MoTLD performs well, but tries to Re-ID, would need to research the library more.
  - Thinking I will implement the tracking stage first, and have each detection that would go to the neural network be a new target for now, to ensure no interference
  - Probably using MoTLD, as comitting to it will give me more time to study how the library works

- In terms of neural network to use:
  - Many online reviews prefer TensorFlow to Caffe
  - Lots recommend using keras as a wrapper around tensorflow, but this is python only
  - What does Toby have experience in, as this seems to be the best way.
  - Don't understand how to design neural networks properly, how to choose number and size of layers. (experimentation)
  - Suggestion from some papers has been a neural network per image to get features, passed into another to tell if the person is the same or different.

Meeting 4

- Let's use MoTLD, try and make it as an abstract class so we could potentially switch tracker, then write multi-object tracker class to inherit from this and call tracker.
- Making tracking system work first priority
- Toby has code on git to convert Mat to IplImage, try and stay as far from this old C stuff as possible.
- Siamese neural network seems like the way to go.
- OpenCV DNN can read TensorFlow and Caffe Models, will use this
- Write abstract class design by next meeting.
- Vague timing plan by next week

Week 4

- Made skeleton in main
- Did OpenCV reinstall, TLD implementation is now faster but performs terribly, so still will not use - check on changes regularly, if I can use it that would be great
- Written abstract class with virtual methods, now writing MoTLD version

- ObjectBox is defined in Matrix.h in MOTLD - understand how this works now and have converted Rectangle
- Having difficulties with compilation with the header file library, multiple definition issues.
- Compiles when the implementation of my tracker is in main.cpp, and tracker.cpp is ignored, still unsure why this is the case.
- Also don't really understand abstract classes in header files properly yet, will continue to look into it.

Timing Plan - as of week 4

- Next 2 weeks to implement the tracking component with placeholder for neural net.
- Spend time here investigating other tracking algorithms if necessary.
- 2 weeks research and experimentation with neural network design, most likely in python with keras, while also looking into forming dataset
- Then start to implement neural network on our dataset and establish the most effective settings for it.

Meeting 5

- Convert MOTLD into non-header only library, and republish on github.
- Need to pull out the patches to allow us to pass the identifications to the neural.
- Check if keras output format is tensorflow, and can be read by opencv dnn module.
- Discuss data collection issues next time.

Week 5

- Converted MOTLD from a c header library to a normal C++ library, and have resolved my linker issues
- Tracker.cpp mostly implemented, but still untested and unused in main.
- Unable to ssh into cameras, α and β are bringing me to a password page, γ seems to not exist

Meeting 6

- Deleting targets function next priority, then integrate with main
- Can now get into camera alpha (in the lab) and beta (on the roof), gamma and delta have lost connections, Toby needs to add my key to these when they come back up.
- try and get onto these cameras and follow instructions on bitbucket

Week 6

- Implemented tracker in main, next thing to do is work out how I'll deal with personIDs.
- Hopefully dealt with personIDs using tuples, implemented a waitKey bodge to replace the Re-ID system for now.
- Creates additional targets for some reason, still looking into this
- Return values of tuples are not working
- Fixed return values of tuples, base tracker functionality now works, having problems with deleting targets.
- Seems to get stuck just before the person leaves the scene, meaning that the box doesn't get close enough to the edge to be deleted.
- Need to associate people objects with tracker objects, will require a rewriting of the person class.
- Deletion function not working properly, currently unclear why.
- Need to make the wait key return the correct numerical value.

Meeting 7

- Investigate potential for use of contrastive loss CNN (Collomosse work).
- Keep going with the deletion logic, last stage of tracking component.

Week 7+8

- Implementing the delete function in tracker
- Changed MOTLD to include deletion functions, still erroring on some stuff, working on this further
- Deletion logic functioning sensibly, but still haven't got it working with deleting the NN classifier or the fern filter.
- Need some logic to delete targets in the middle of the scene (not near the edge)
- Current logic works well if the first detection of the target is a good one
- Boxes don't always get close enough to the edge of the image to be deleted, maybe have some logic in the re-id part to delete target if it clearly is nothing like the target it is supposed to be.
- Deletion heuristic (how far away from side of the image) still needs more work/research.
- Thinking the best idea will be to keep the edge of the image heuristic, and also have one to delete the object if it is not similar enough to the saved profile.
- Ferns deletion still needs to be done, will be quite complex to do by personid as this is not currently used for this piece of functionality.

Meeting 8

- Discard current MoG and Haar
  - when something enters, gets generic type object (set colour)
  - if some distance away from side, run HoG on it (only once, or up to N times) (give person colour)
  - Then let MoTLD take over
  - Maybe when it gets close to edge, run HoG to check if still a person, if not, delete
- Plan out CNN architecture - base it on transfer learning from Collomose
- Do above changes to the MoG and HoG system
- Try and understand Ferns code in MoTLD

Week 9

- New Pseudocode:
  - Adding TLD targets
    - Run MoG on every frame, and draw generic colour box
    - If away from edge and not currently in a TLD box, run HoG/Haar
    - If HoG/Haar successful, add TLD object
  - Deleting TLD targets
    - If leaving the edge of the image
    - If stationary, run HoG/Haar
    - If HoG/Haar unsuccessful, then delete
- Also need to properly understand the Ferns code to correct my deletion function and get rid of the enumeration errors
- Changed the adding target logic, deletion of NN objects causing a segfault when multiple targets are present
- Haar does not give a target in many of the MoTLD objects.
- Changed to HoG, has a much better hit rate in these detections.
- The above pseudocode has now been implemented.
- Cannot handle deleting two targets in the same frame, need to look into this
- Notes on what is wring written on paper, as well as very brief CNN design.

Meeting 9

- Have another new paper to read
- Need to fix Fern problems, only thing left now in tracking as far as I know at this time

Week 10

- Fern problem now fixed (mostly), may investigate further to try and improve quality of TLD tracking.
- Need to fiddle with HoG to get more reliable detections.
- Now need to associate targets with person objects - add function to merge TLD objects
- Keras had a siamese NN example
- Look at papers for structure of other networks for re-id or feature extraction
- Possible dataset in download, don't know how to open rosbag files though.
- Nice network architecture on page 9 of "Embedding Deep Metric for Person Re-identification"

Meeting 10

- Need to make another version of tracking code to extract training data for my network
- Will actually be trained from the cameras on the roof, but the data I will get for now will serve to test the implementation.
- Toby is finding me some data

Christmas TODO List

- Integrate person class and add merge function
- Make data extraction class (to get training data from videos)
- Implement siamese CNN using keras example and network architecture from papers
- More reading on siamese networks for this purpose and feature extraction network architectures.

Christmas Notes

- Keras installed with tensorflow GPU backend - had to downgrade CUDA to make work, didn't need to do CudNN
- Created dataExtraction file, need to make it put the images in a nice format.
- Data extraction working, but TLD slow to track targets moving sideways, and HoG is not getting good results - original HoG example from Toby is not better
- Experimenting with the Siamese network using the MNIST dataset, the simpler network seems to work best, need my own data now really.

- Made the way i deal with rectangles and personIDs much nicer, still don't use the person class, passing it as an argument means that it would have to return the updated tracker, which is a waste of time.
- Think I'll only use the person class for colour/identifier and patch extraction from the MOTLD object using personID.
- Need to update data extraction class to work with these changes - DONE THIS NOW
- Read a few more papers on siamese CNNs
- Need to resize the images before sending them to the neural network, and in the data extraction class.Have set this to (128,256)
- Have several NN files - the original example (exampleSiameseNetwork.py), siameseNetwork.py, which will have parameters for my dataset, and networkExperimentation.py, which is being used to help me learn keras better.
- Network experimentation is now in a good spot, need more of my data now
- Data is downloading from Toby's server, once I have it need to do a lot of supervised downloading. Also need to establish if the same people are present in multiple videos and label them accordingly.
- Got data, lots of videos have no people in them so currently in the process of removing them. Whole gamma folder is useless so has been deleted.
- HoG person detector getting nothing from cameras on the roof, need different approach.
- Haar detections don't occur every frame so would require constant NN classification
- MOTLD does not follow targets moving directly sideways accross camera view very well
- Got HoG to find people in the roof camera videos by enlarging the image
- HoG gets weird detections on the alpha camera of my dataset from last year
- Improved organisation of images extracted
- HoG call is causing the segfaults
- Fixed issues by increasing the size required for HoG detections
- Also rectangles that were within each other were both being added and targets, have now fixed this.
- Deleting target 0 before target 1 causes a segmentation fault, am investigating.
- Need to change my deletion function to work off actual personIDs in both dataExtraction and main.
- Have now altered the delete function so that it works correctly, seg fault no longer occurs in this situation
- have made bash script to remove all data in dataset while testing
- extracted and filtered dataset1 data

- Ensure smaller people inside someone elses MOTLD box doesn't result in the target being kept, maybe introduce a size threshold - DONE THIS NOW
- Still having problems with HoG, particularly in alpha camera when Caz is in view, and moving sideways
- Dataset 2 not good for data extraction as too many people are overlapping.

Meeting 11

- Dataset of people should be something like an ordered heap based on when/where last seen, and queried in order, stopping when the confidence level is high enough. This is a measure to increase the scalability.
- There is an unlabelled thermal pedestrian dataset if we want a lot of negative examples.
- Next objective is functionality to read in images and give to keras.

Week 1 1

- Size threshold currently set to 2, causes a lot of deletions, experiment with this.
- Neural Network occasionally errors for currently unknown reason after calculation of accuracy, am investigating. Is an ExceptionIgnored error, and seems to have no effect, but I still don't like it.
- Need to investigate MOTLD process frame call to stop it drawing boxes so I can draw boxes and write text of my own colour choice.
- HOG is still causing me a major problem with erroneous detections, as well as MOTLD not moving sideways across the frame very well
- Did data extraction on all of last years data.
- Reading in data to network, issue now is that I am running out of GPU memory.

Week 12 and Meeting 12

- Experimenting with generators, nearly there I think.
- Now seems to work but will need ages to run it
- Do other stuff nice way of outputting the model - save model, weights, parameters, confusion matrix, precision, accuracy, recall
- Shrinking batch size fixes graphics memory error
- Dataset now working correctly with generator, not sure if the evaluation is working properly, need to test on mnist

- Toby is sending me login for compute servers
- While filtering the dataset, found more missclassifications than I expected, need to check data extraction class, maybe MOTLD is doing erroneous re-ID
- Connected to NCC and install keras and tensorflow, need to use “module load cuda/8.0-cudnn6”. Have cloned my repo on to it, and used scp to put my data onto it. Now need to write appropriate slurm scripts.
- Weird prints when training are allocation fails

Week 13 and Meeting 13

- NEED TO FILTER DATASET AGAIN - DONE
- Been fiddling with train test split, need separate data for training, validation and evaluation. Split into training and validation, maybe use a different dataset for evaluation.
- Reference survey paper for first part of lit review
- Don’t rely too heavily on last years paper for this year, as it will be different marker
- currently almost no learning is happening, need to look at this.
- Testing a fair amount of networks from the internet - none worked so far
- Implemented a closed world architecture for my data, this appears to be working well.
- Closed world with augmentation - Test accuracy: 0.9645250295624753

Week 14 and Meeting 14

- Try replicating my generator logic for the MNIST dataset and see if that will work - done this now, it is running on NCC
- Also try downscaling alll my images massively and loading into graphics memory to see if this will work.
- Experiment with parameters around learning rate and check effect on loss.
- The generator is the problem, tested on MNIST and getting 50% accuracy.
- Testing with my create pairs fucntion, didn’t work, think problem is with my logical comprehension of dataset.
- Have it working now when all loaded into video memory and not using generator, need a pascal for this though.
- Need to define a new accuracy metric, otherwise the graphs aren’t useful and I’d like them in my paper
- When model is saved, its output will be the difference in vector space rather than a label of 1 or 0. Current confidence threshold is 0.5.

- * Accuracy on training set: 100.00% * Accuracy on test set: 97.85%
- Need to make new evaluation dataset

Week 15 and Meeting 15

- Trying to use K.cast and K.less to have the euclidean distance return 1 or 0 instead of the distance to make the accuracy measure work, but having issues with None values. This is apparently caused because tensorflow represents zero as none.
- Changing accuracy metric to binary doesn't work
- RMS optimiser breaks everything
- Changed threshold to 0.3, accuracy now 99.35%
- Threshold at 0.2 gives 100% accuracy, getting slightly suspicious now of how good this is.
- Make a little script to learn about the OpenCV DNN model by randomly loading images from the dataset and passing them to the network and get the distance back. Will also help to test my suspicions about the 100% accuracy.
- I've put the euclidean distances from the predictions in a text file, seems to back up the 100% accuracy.
- Meeting
  - Define evaluation dataset – ongoing - use Toby and Miko from behaviour dataset
  - Define new accuracy function for TensorFlow.
  - Re-run and evaluate CNN model inside OpenCV tracking run-time. - make toy example first, then integrate into my tracking code
  - Consider capture of performance metrics (inc. ROC curve + distribution of embedding space).
- Made new accuracy function, fairly happy with it although still needs more testing.
- Can't do ROC curve in tensorboad, will be done in sklearn and matplotlib i guess.
- Gathering evaluation data, HoG not performing well close to camera, maybe have an if statement to check for larger targets first, then look for smaller.
- Current implementation will be good from the roof cameras.
- Gathered evaluation data - Toby and Miko saved in 8 and 9.

Week 16 and Meeting 16

- ROC curve from distances, get TP and FP from here
- Do a “grid search” of different threshold
- Include top 5 error in paper if time.
- Best way to save my trained model is with an h5 file
- Need to ensure I have the latest version of opencv and contrib
- People seem to be having issues reading in networks with Flatten layers, will investigate.
- Having OpenCV issues with cuda - not rebuilt yet and still works, maybe don’t need a new copy
- Think I’ll have to convert model into a tensorflow .pb file as no one seems to have done it with h5.
- pb conversion didn’t work with my custom loss function

Week 17 and Meeting 17

- Had to specify my custom functions in the conversion file but have successfully converted to pb. Had to do this on the NCC for graphics memory reasons.
- Model currently saved locally was only over 10 epochs
- Now working on ROC curve
- When checking thresholds, change the tensorboard names so can use these graphs to express results.
- Also need to test number of epochs, is 100 necessary??
- Drawn ROC curve, is dumb and pointless - also seems to be backwards?
- Having a weird issue where my seeded random generator
- NEED TO CONFIRM WHAT DATA TO USE ROC CURVE WITH
- Meeting
  - Try ROC curve as top 5 error
  - Embed high dimensional data using t-SNE - will be useful for poster and paper
- Replaced flatten layer with tf stuff, training worked, yet to try with DNN
- Removed early stopping, don’t like it
- DNN still failing, fairly convinced I need fresh install
- Finally managed to rebuild OpenCV, failed at 100% on the contrib modules though

Week 18 and Meeting 18

- Tried lots of stuff, dnn still not working
- C++ not the problem, python also doesn't work
- Flatten error fixed, this was well documented, Maxpooling now giving error, can't find people having this error before
- Saving as tensorflow straight from keras script made different error -
- Alternatives
  - Could run Neutral Net in a separate thread and pass pairs of images to it
  - Or rewrite the whole of main in python and have them run together with using DNN
  - Or possibly use C++ bindings of tensorflow
- Skip DNN and use C++ tensorflow
- Maybe compare using TLD to just calling the neural network every frame
- Successfully built C++ tensorflow
- For C++ Tensorflow:
  - Need to symlink my src to tensorflow/cc/ –¿ ln -s src/ /home/tom/tensorflow/tensorflow/cc
  - Need to have a .cc file and a build file in my src directory, to run with bazel -(bazel run -c opt //tensorflow/cc/example:example)
- Another alternative - nest python in C++, create python file that reads in network and then has a fucntion with 2 images as parameters which returns the distance.
  - for this, need callModel.py, which loads the model (globally) and takes 2 images in from the command line ??
  - Would then have this return the confidence level
  - C++ would use second answer from https://stackoverflow.com/questions/16962430/calling-python-script-from-c-and-using-its-output

Week 19 and Meeting 19

- Made python re-id system
- Network predictions are all zero, unsure of problem, retraining now as unsure what version of the network I have
- mkfifo - investigate running a python script separately
- Test out ROS code on cameras.
- Camera Gamma is currently offline, go on without it for now

- 10 is a bit slow when the number of targets increases
- change logic for saving frames - maybe every 10 recognitions?
- Re-ID looks like it could work well on my new dataset, need to do experiments with the threshold though. Experiment and record loss and accuracy for all params from 0.1 - 1.0

Easter Holidays

- Need to grid search network params and classification threshold
- Made progress on extracting intermediate feature vector, many of them are zero......
- SKlearn grid search doesn't work with the siamese network for some reason.... should now try hyperas - also didn't work, will investigate further tomorrow.
- Hyperas grid search now running, think I have a way to extract intermediate layers, will test when optimal params are found
- Writing test script for mkfifo
- mkfifo testing script worked
- Integrated mkfifo logic into main and new nnUtils files, currently untested.
- Grid search ran, results in bestparams.txt
- Toby sending feedback from last year
- Haven't arranged another meeting, if I need to discuss something email Toby and arrange a time.
- C++ and python mkfifo functionality now works
- This means I have TLD back to talk about!!!!!!
- USE NVIDIA-SMI TO RESET GRAPHICS ERRORS
  - this is kinda annoying, maybe do a CUDA, CUDNN and tensorflow reinstall if time
- Network has been retrained and updated
- Not sure labels/rectangles are going onto correct people - problems with TLD object enumeration again
- Try averaging classifications over all known images of a person and check if below threshold.
- Fixed enumeration
- Next
  - Change Re-ID thresholds - maybe re-id if confidence is much greater than other targets.
  - enable multithreading and mutexes like last year

        - ∗ will need to investigate if they can all share the same fifo
    - determine correct number of frames to save
    - restructure and use this year's dataset

- Old network had better graph on the training data, but was overfitting7
- What to prioritise, no misclassifications or minimal number of targets. TALK ABOUT THIS CHOICE IN THE PAPER
    - if re-ided, then defininetly that person, but may create new target - I THINK THIS IS PREFERABLE.
    - OR, create fewer targets, but misidentify
- MENTION LACK OF DETAIL AVAILABLE FROM THERMAL RECORDINGS
- Current best network
    - * Accuracy on training set: 99.93%
    - * Accuracy on test set: 97.62%
- 6 new papers to add to related work
- TRANSFER LEARNING FROM CLOSED WORLD???
- we don't have to require all positive pairs holding smaller distances than all negative pairs regardless of query images. Instead, what we want is for each query image, its positive pairs have smaller distances than its negative ones. - THINK ABOUT THIS
- Person objects are clones of each other, unknown why - fixed now
- Transfer learning results weren't great, grid searching the closed world section now
- Video writer problems are to do with the way my results videos are output, original videos can be combined fine.
- Sleep is wrong, each iteration needs to take a set amount of time - done now
- Cameras gamma and beta have overlapping FOVs
    - remove activity condition??
    - crop out one of the FOVs??
    - delay start of gamma (video editing) to ensure boudary, will also have less time between leaving gamma and arriving in alpha.
- TLD targets appear to be jumping between people.
    - zero jumped to wrong person for a couple of frames, new tracker object created in its place, jumped back...
- Old python rewrite not showing promising things for the network
- NEW TO DO

  - sync up data, weird things have happened, written on paper
  - network optimisation
  - re-id optimisation (threshold, how many to save etc.) - THIS WILL BE IMPORTANT, NETWORK ACCURACY IS ALREADY PRETTY GOOD
  - Video merging code

- Transfer
  - * Accuracy on training set: 99.52%
  - * Accuracy on test set: 94.68%

- Retraining current best architecture (non transfer with simple augs)

- fixed tld issue with disappearing targets (i think)

- fixed memory leak and post processing problems

- coming to the end of network experimentation

- CONCAT AND SAVE POSITIVE AND NEGATIVE IMAGE PAIRS FROM RE-ID SYSTEM TO PRESENT - WRITE DISTANCE ON THE IMAGE, MAKE THESE INTO VIDEO
  - also useful to debug re-id system

- Get outliers from training as figures

- NETWORK HAS OVERFITTED TO TRAIN DATA, THIS IS THE PROBLEM WITH RE-ID SYSTEM
  - running new grid search to optimise for eval data
  - maybe try cutting beginning and end off eval files, these seem to be the most difficult bits

- mitigating overfitting
  - more dropout - experimenting in grid search
  - reduce network complexity, experimenting now
  - regularisation
  - more data (augmentation)

- reg - 0.0001 ——¿ * Accuracy on training set: 97.07% * Accuracy on test set: 95.57% * Accuracy on eval set: 69.06%

- when trained on new data, get * Accuracy on training set: 90.52% * Accuracy on test set: 89.57% * Accuracy on eval set(same set): 90.31% - data is doable

- TRY TRAINING ON TRAIN+TEST AND VAL ON EVAL

- "I almost hesitate to call it over-fitting because sometimes the test results are acceptable" - some bloke on Stack Overflow

- Validation set does not decrease, so not overfitting that much
- List of tests - run separately
  - full aug + simple net
  - full aug + full net
  - dropout after every layer
  - larger layers - 128
  - larger layers - 256
  - larger layers - 512
  - l1 reg
  - Batch Normalisation - RUNNING NOW - not good, changed axis
  - reduce batch size
  - higher droput rates (90%)
- SHALL I RECORD ANOTHER ROOFTOP DATASET AND USE THE EXISTING ONE TO TRAIN AND THAT ONE TO EVAL?
- Am I overfitting?
- Train 99, val 96, eval 70(ish)
- val not decreasing at end of training
- 70 still better than random
- CAN I ZOOM THE THERMAL CAMERAS?
- FOR THE TRAIN
  - currently have new predictions and new model locally, update spreadsheet and try full re-id
- ################################
- New network does not plateau like the old network, will need more epochs
- Re-ID system works fairly well, missclassifications seem to be due to hog classifications smaller than a person
- Grid search ran out of memory
- HoG:NN
  - Alpha 1259:65
  - Beta 1863:114
  - Gamma 2454:339
  - This ˆ is the reason for speed difference - everything in a mutex??
- Toby will read from results onwards, highlight anything I want him to read beforehand