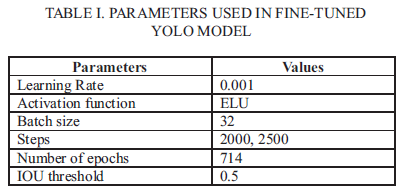
**IR detection:**

20-08-(Yolov3) Real time target detection for infrared images:

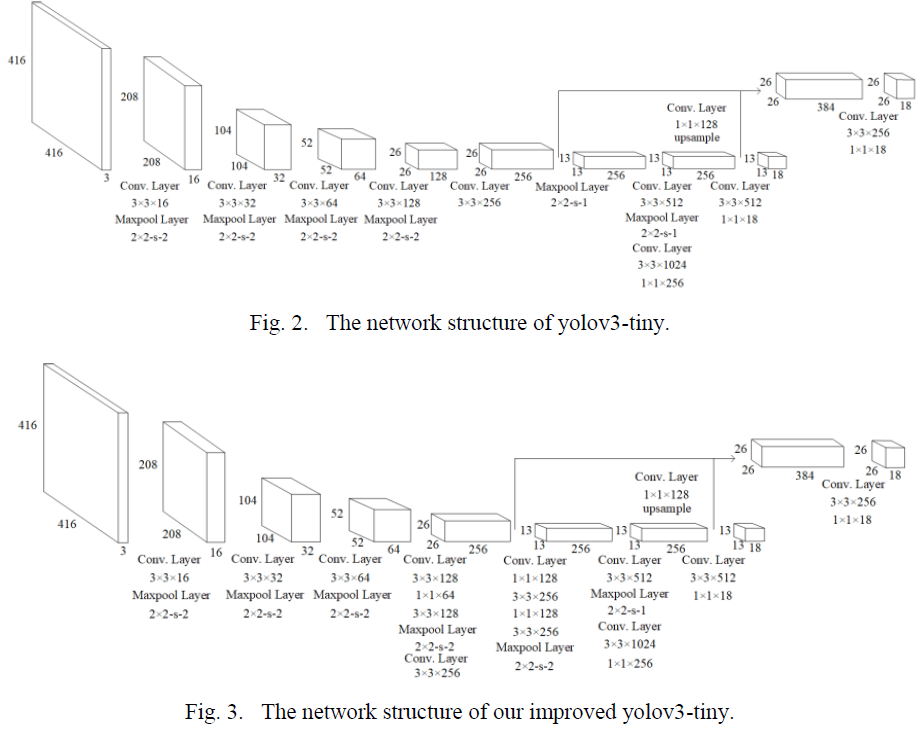
* **Training data is normalized [-1, 1]**
* Maybe we run Kmeans for anchor boxes of IR?
* But pay attention that Euclidean distance is not suitable for yolov3 since it produce more error for large boxes and less for small ones



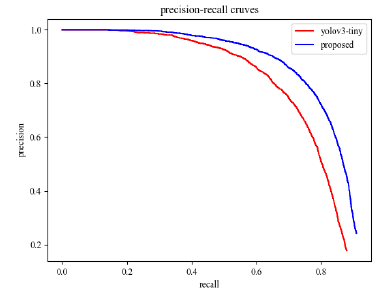
20-07-(Yolov3Tiny) Vehicle detection in thermal images with an Yolov3Tiny**:**

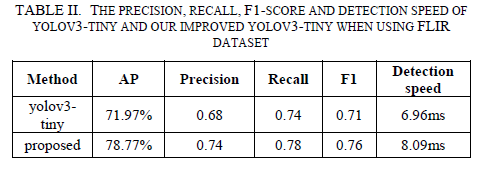
* **Two major improvements has been done. 1) k-means over the dataset for new BB 2) Deeper structure w.r.t. yoloTiny for better feature extraction of vehicles with higher acc.**
* **For K-means use the right distance function and not Euclidean**





* The specific improvement is to add a serie of 3\*3 and 1\*1 Conv. Layers to the original structure of yolov3Tiny.
* 9 anchors per grid is used for better tradeoff btw. Computational complexity and recall
* Read the paper for more info. about training parameters
* 78.77% mAP on FLIR

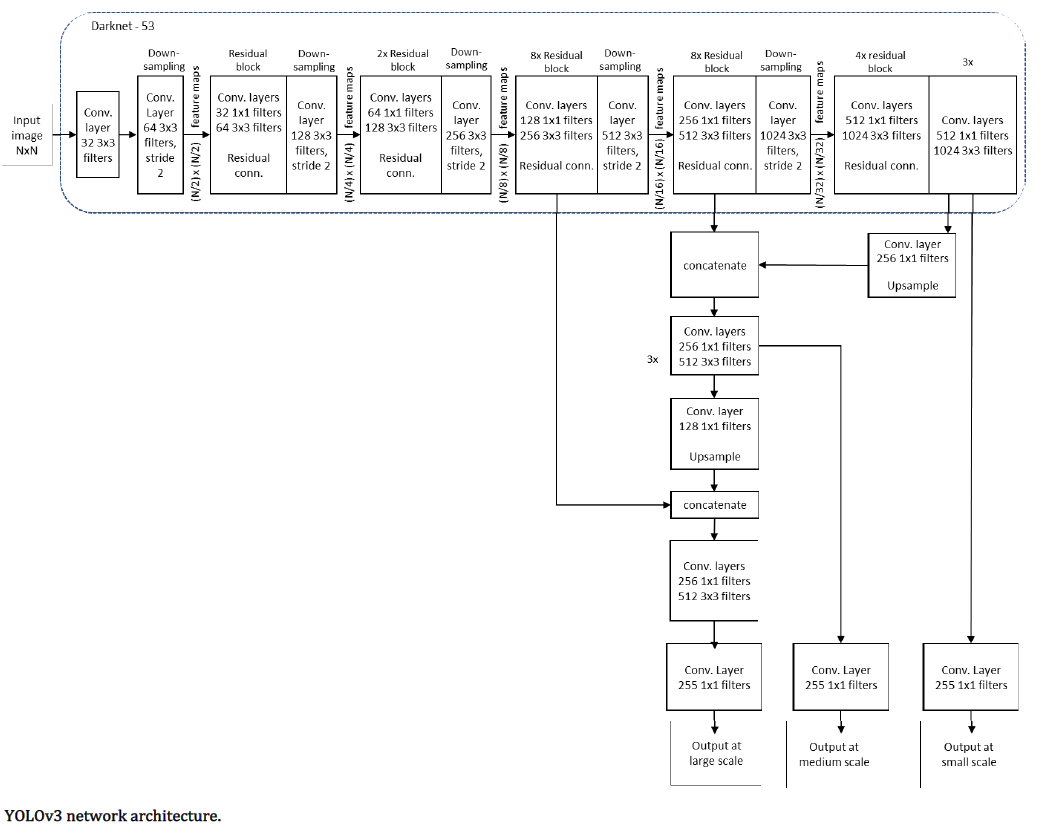




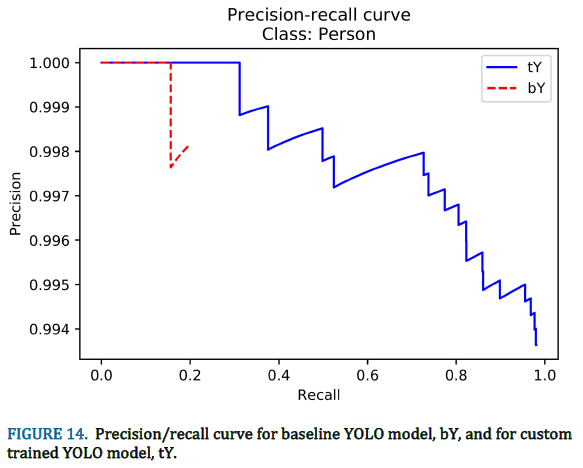
* Detection accuracy is not enough but it is fast!

20-07-(Yolov3) Thermal obj. detection in difficult weather**:**

* Yolov3 is faster than R-CNN, SSD and Cascade R-CNN with comparable AP



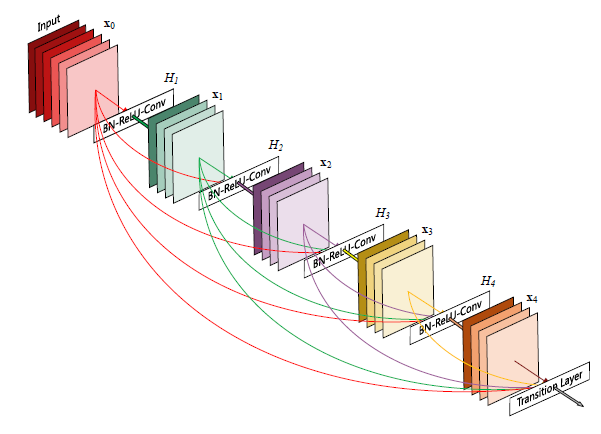
* Yolov3 pre-trained on COCO is used as baseline model → name it bY
* Baseline plus 4270 IR images training → name it tY
* For training details read page 9 of the paper
* Evaluation process is explained well, might be needed later
* Detection is compare to ground truth and is true positive if IoU > 50%

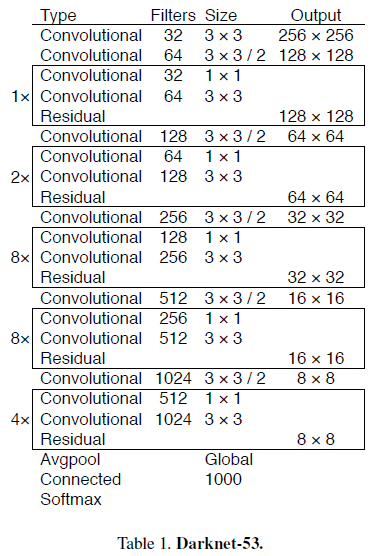


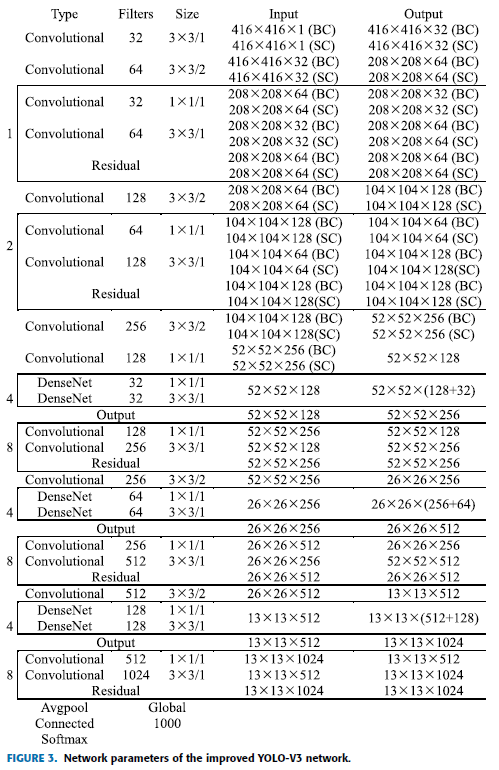
* Precision-recall curve for a class, provide the AP
* Original Yolo on person → AP 19.63% with 15.5% recall at 100% precision
* Yolo with 3k IR image trained on person → 97.93% AP
* IR performs better in rain since the temperature difference is higher
* They demonstrate that Yolov3 can learn with relatively small dataset (1k) and small number of iteration (1600)
* Trained Yolo shows good generalization properties w.r.t non seen images

20-04-(Yolov3)Using deep learning in infrared images to enable human gesture recognition for autonomous vehicles:

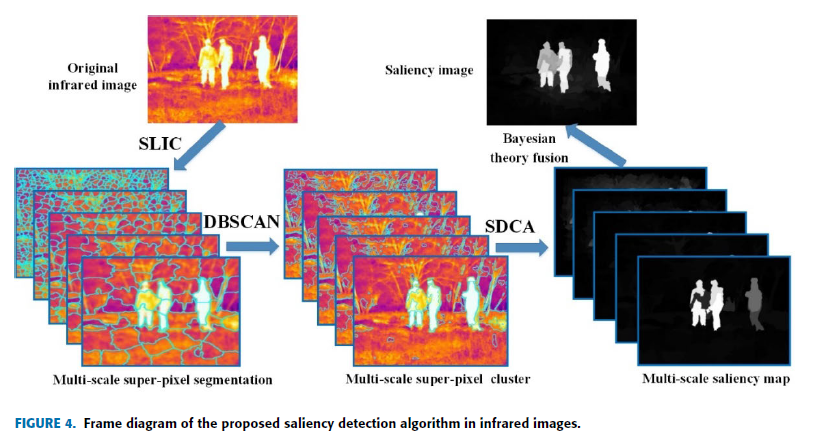
* The network take two inputs. IR frame and its corresponding saliency map to enhance the reuse of features
* Three DenseNet blocks are added before the residual components in Yolov3 to enhance the convolution feature propagation (i.e. feature reuse)
* The feature map from two inputs are concatenated followed by a 1\*1 convolution to linearly merge the features.
* In the process of Yolov3 a large number of feature information is lost and since a target object in IR provide relatively less number of features, it is unfavorable
* To address this issue DenseNet blocks is used before the residual blocks of Darknet-53





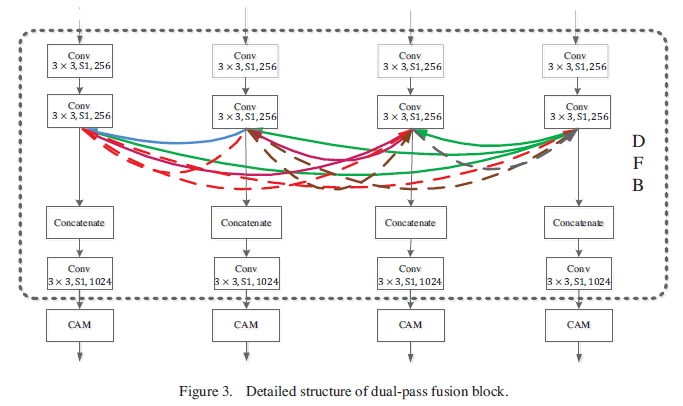


* Due to lack of information in IR frame as well as its high sensitivity to temperature changes, saliency maps is used as the second input
* Read the paper for exact process of saliency map generation

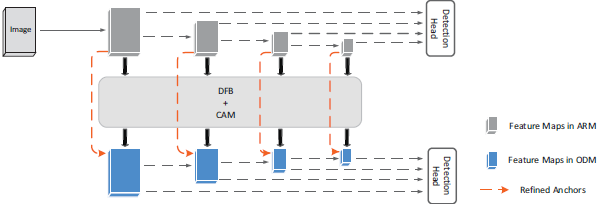


**19-11-Every feature counts an improved One-Stage Detector in thermal imagery:**

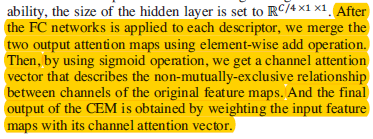
* RefineDet is inherited and further improved to extract better features out of IR frames
* **DFB** (dual pass fusion block) to directly fuse features from all levels
* Then **CEM** (channel-wise enhance module) to adaptively assign weights to different channels of the feature maps
* FPN (feature pyramid network) is fused with extra low level feature maps with different sizes to help the feature quality. Then CEM is used which results in a performance boost

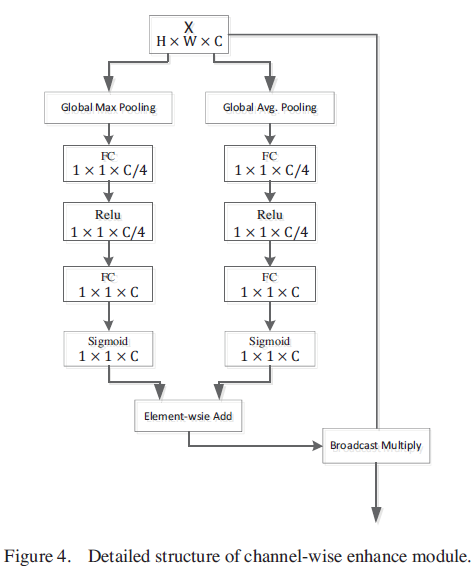


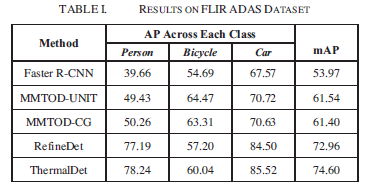
* The structure of ThermalDet inherits the pseudo-two stage optimization method by keeping the **ARM** (anchor refinement module) and **ODM** (object detection module)
* ARM plays a role like RPN to coarsely adjust the size and positions of preset BB and ODM takes the refined Boxes and input to regress the accurate object location, size and label



* RefineDet achieves a great trade-off between accuracy and speed.
* In ThermalDet they use DFB instead of **TCB** (transfer connection block) which is used in original RefineDet
* **DFB:**
  + **Low-level features are more important for images with simple appearance (IR frames). Thus, DFB is designed to take the most of low level and high level features**
  + **In RefineDet the high level features are up-sampled and recursively fused with shallow layers to boost the performance of the small targets up**
  + In ThermalDet:
    1. It is done both ways. i.e. low level is fused to high level and high level is fused to low level
    2. Fusion is by concatenation where, the original method use element-wise addition
    3. All features are concatenated. Low level features with high resolution are down-sampled and deeper features are up-sampled
* CEM:
  + DFB, which contains both low and high level semantical information will be used as input for CEM. CEM’s task is to convert the form to the form required by ODM. The network has to be sensitive to those features of different levels. In other words, CEM will adaptively assign different weights to different channels so that they can be exploited by subsequent transformations most effectively during converting.
  + Channel enhance maps are generated in two different branches after a max/global avg. pool

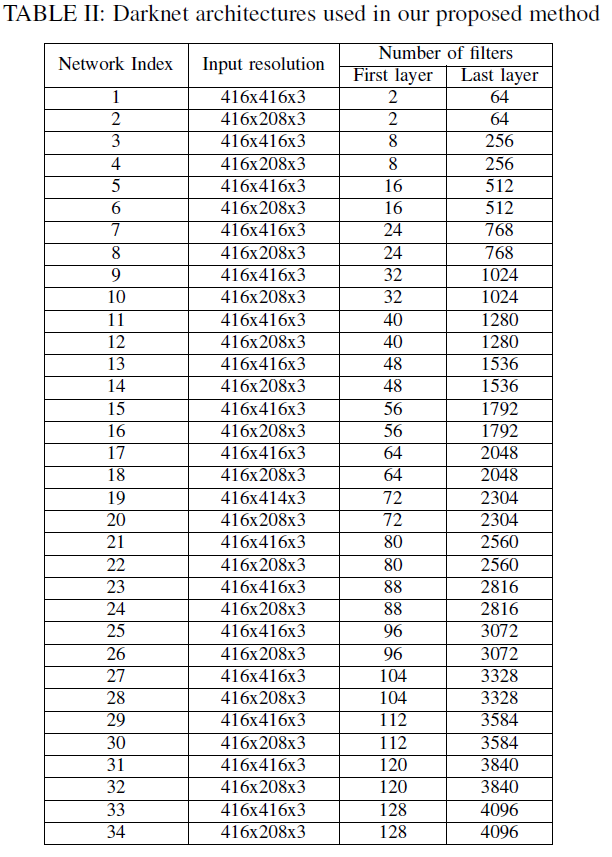




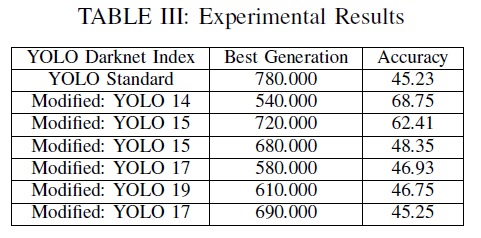


18-12-(Yolov2)Obj. recognition on long range thermal image using state of the art dnn:

* They have tried 34 Networks with different input output size etc.



* Dataset is one million big and is owned for security purposes
* In total 2720 different setting led to 2720 different results. The best is shown below for 8 classes such as Human, boat, vehicle and animal in 2 size of tiny and normal



* Generation is referred to epochs, I suppose