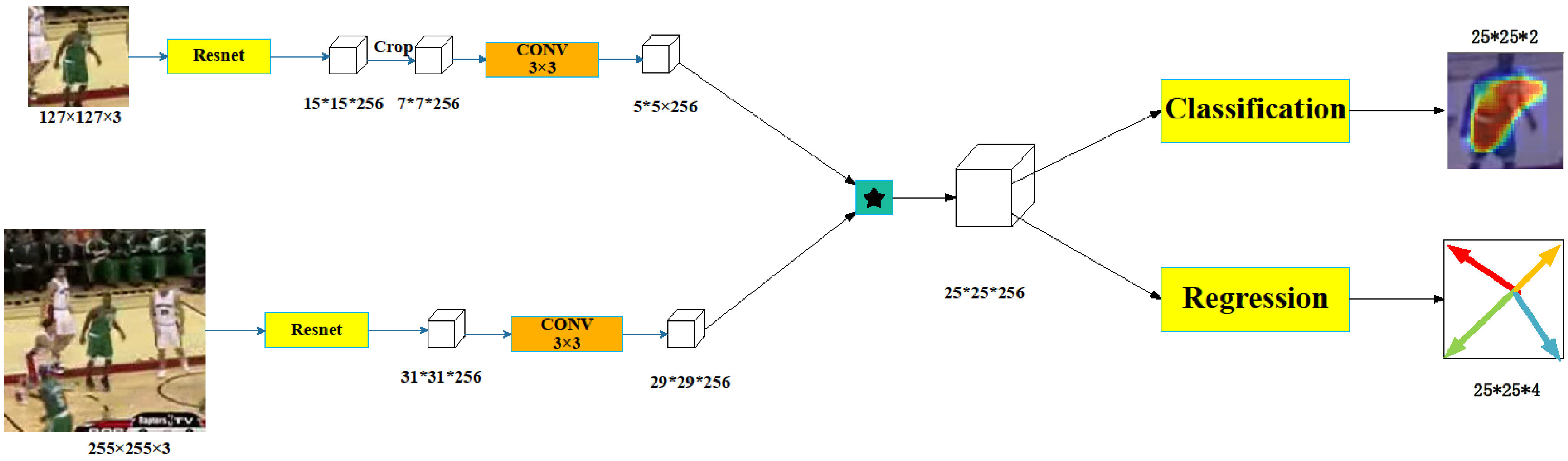
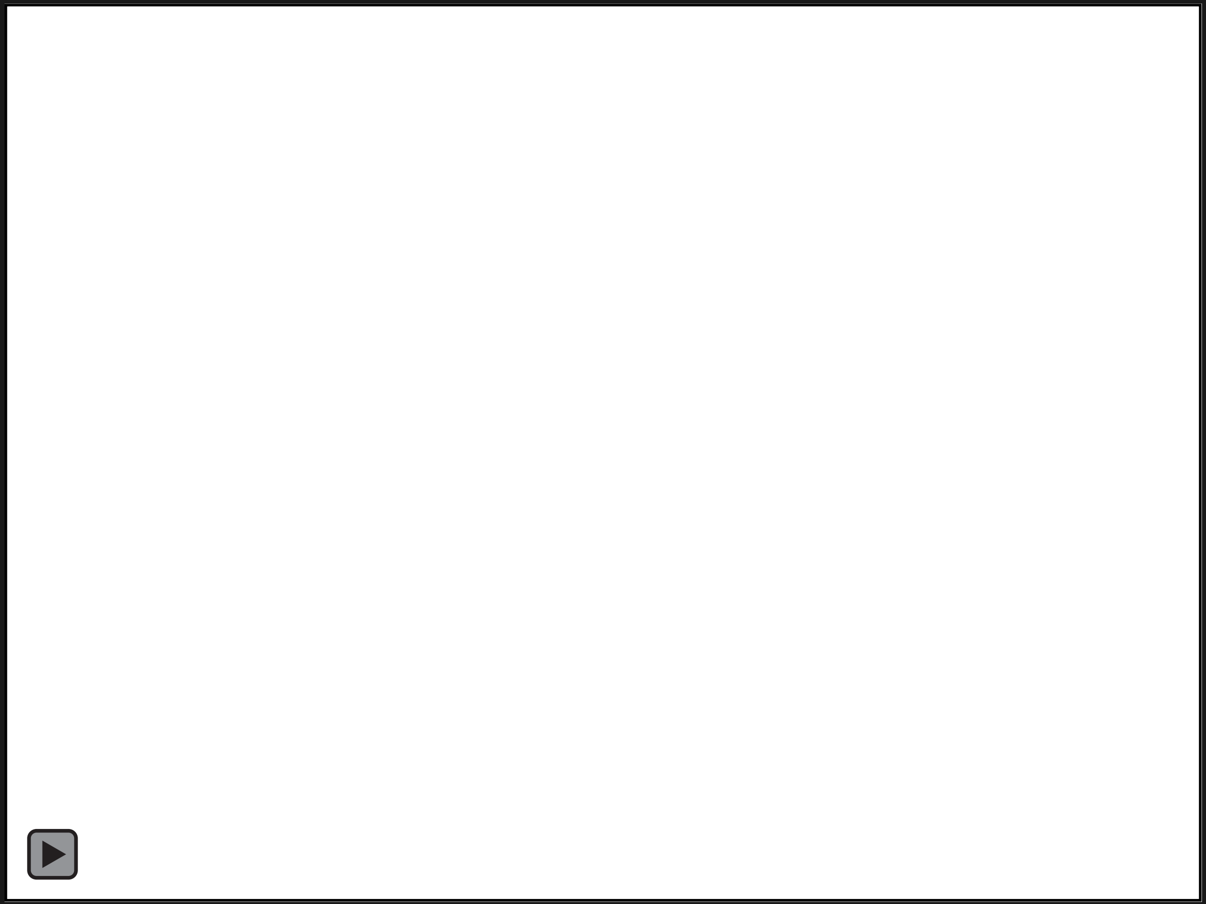
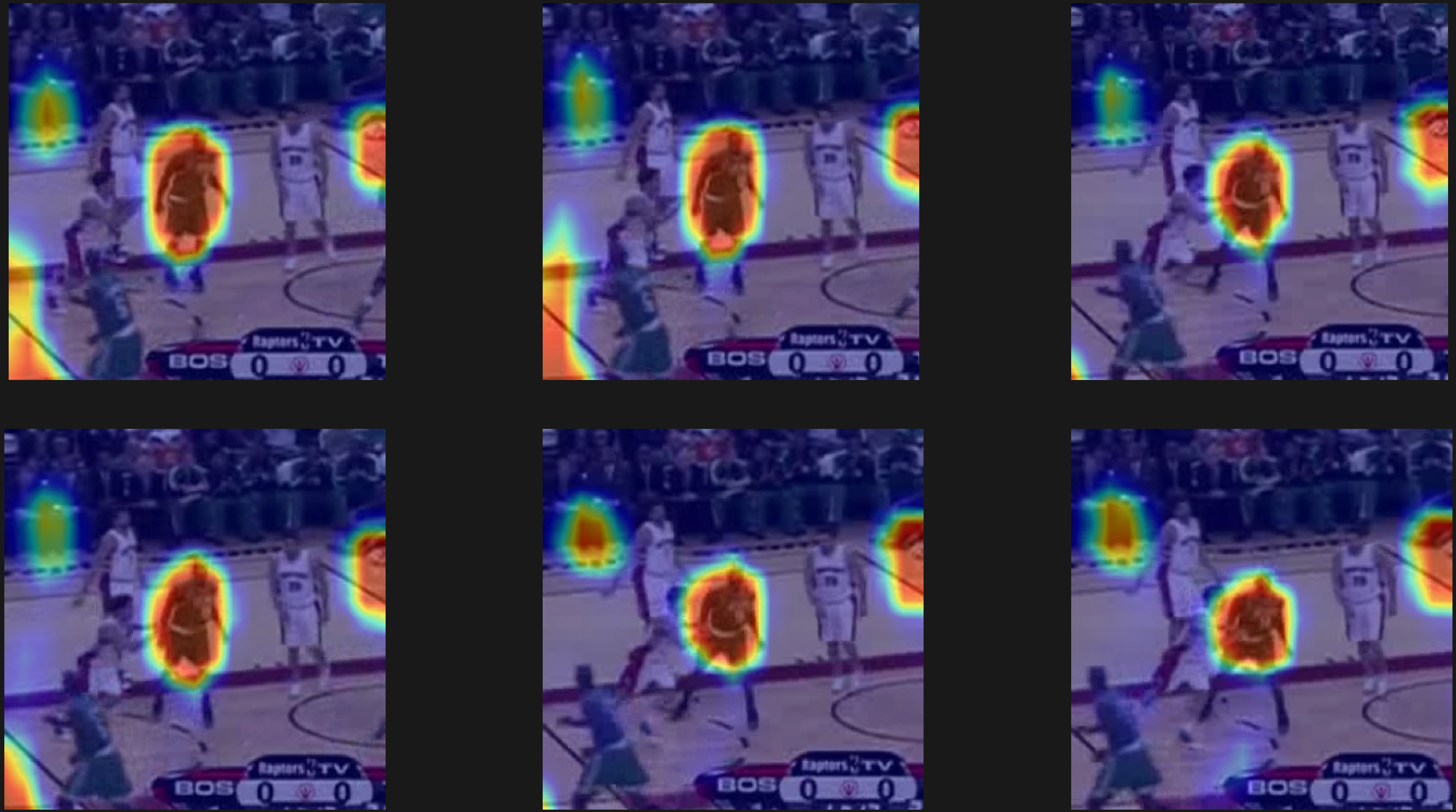
Architecture





Too many high light candidate



New Observation

1, Classification is not supportive
2, Correlation results is wrongly aligned

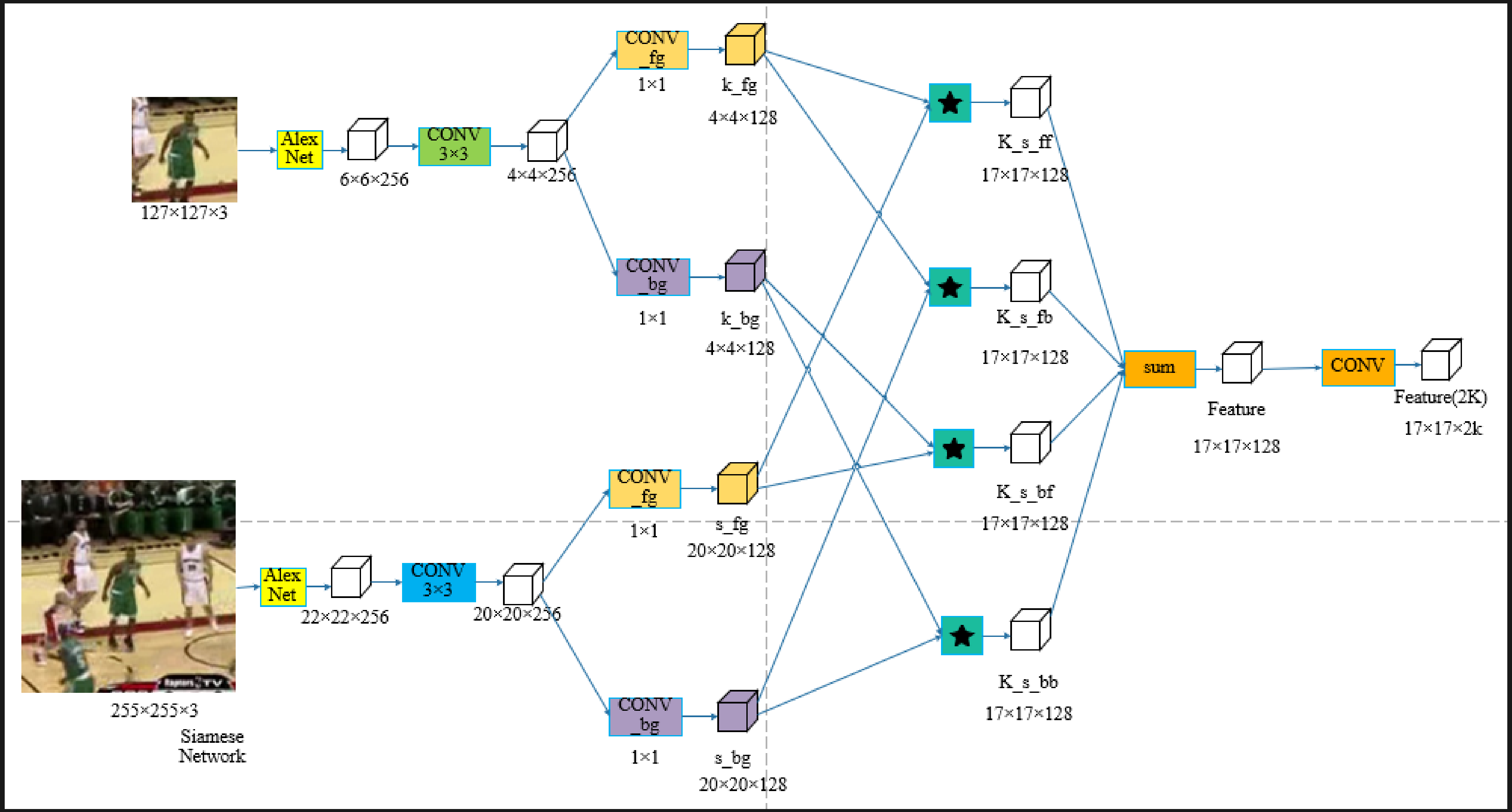
Reason

1, ambiguous ground truth where IOU>0.8 is positive and IOU<0.5 is negative
2, padding with 0 and sub-sampling, correlation realization

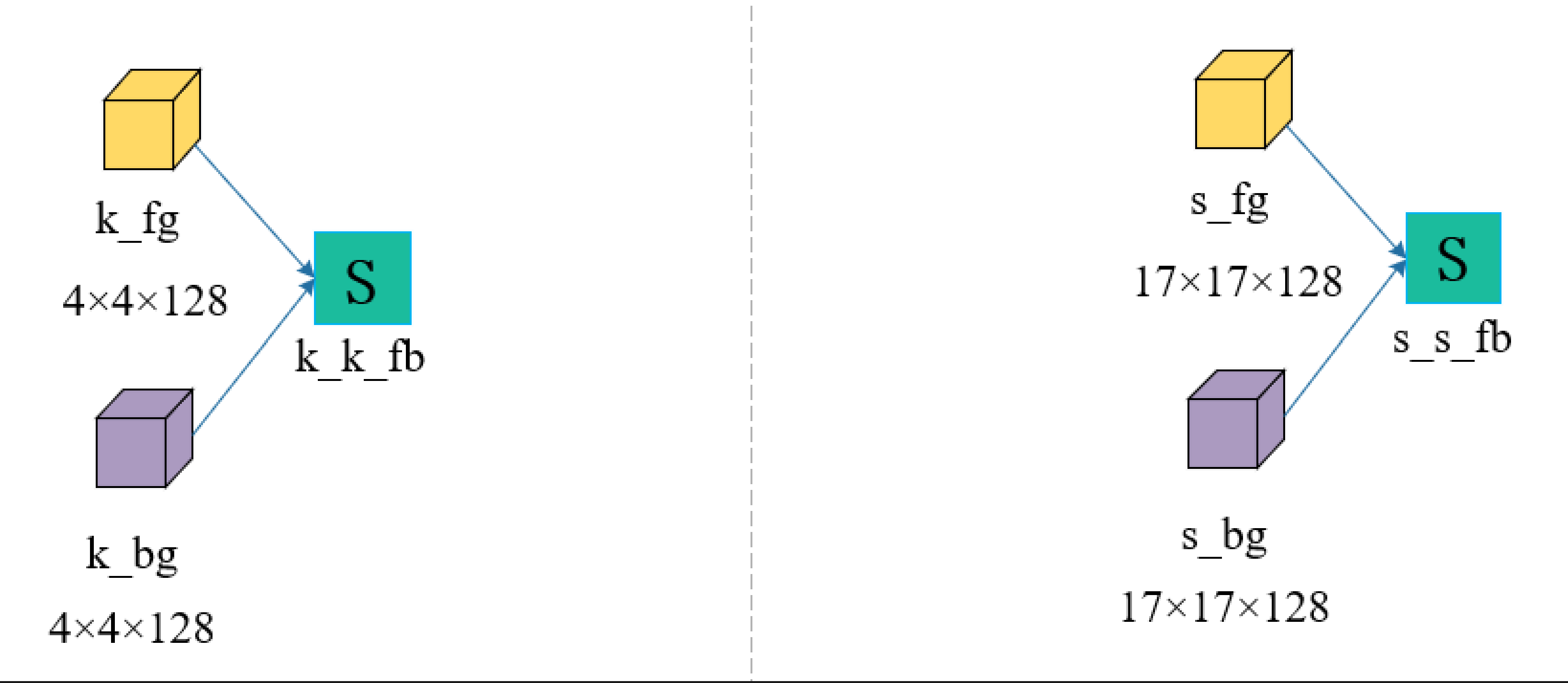
Solutions

1, IOU score, modify the ground truth form
2, Align by convolutional kernel size and padding the results

Disentangle model

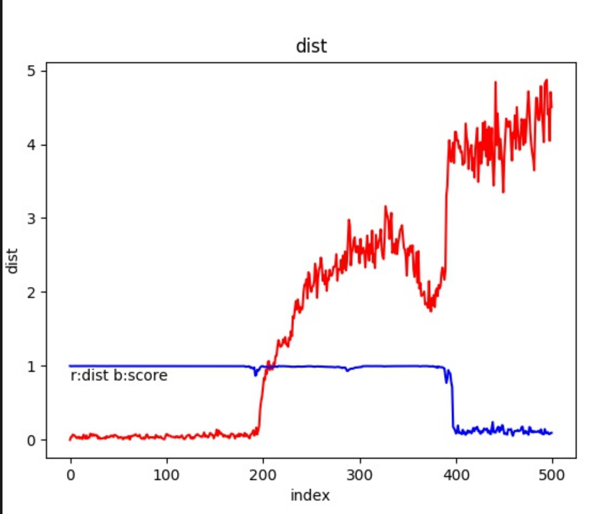


First , the extracted features are split into foreground features and background features to explore the relationship of them.

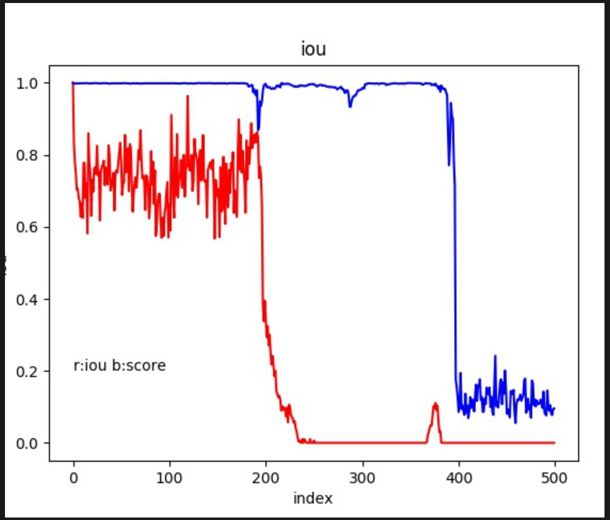


Ensure the separation of foreground and background by introducing similarity between foreground and background into the loss function.

Disentangle model

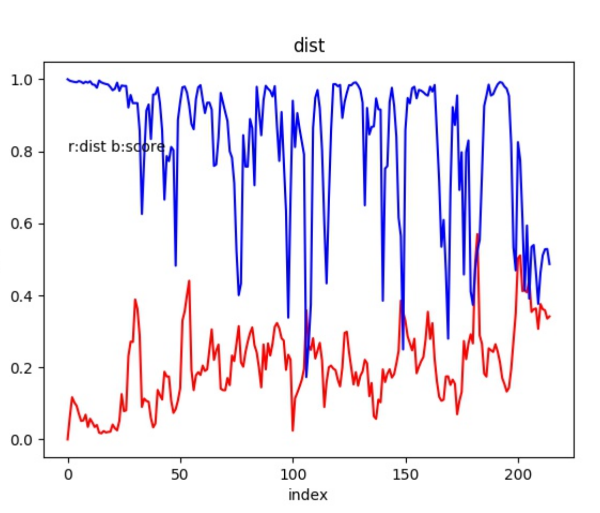


R:dist B： score

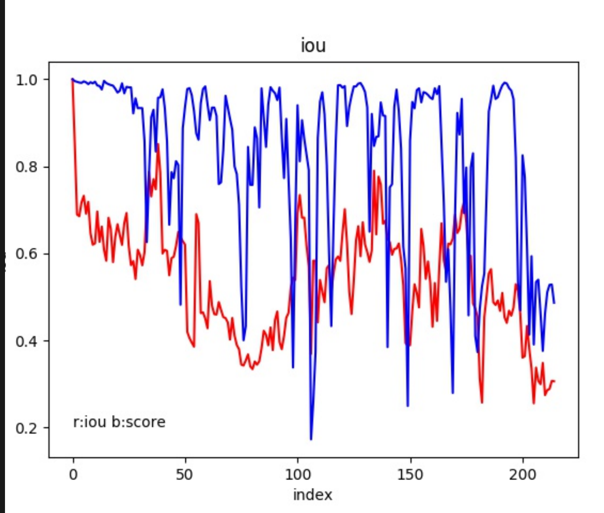


R:iou B:score

Disentangle model

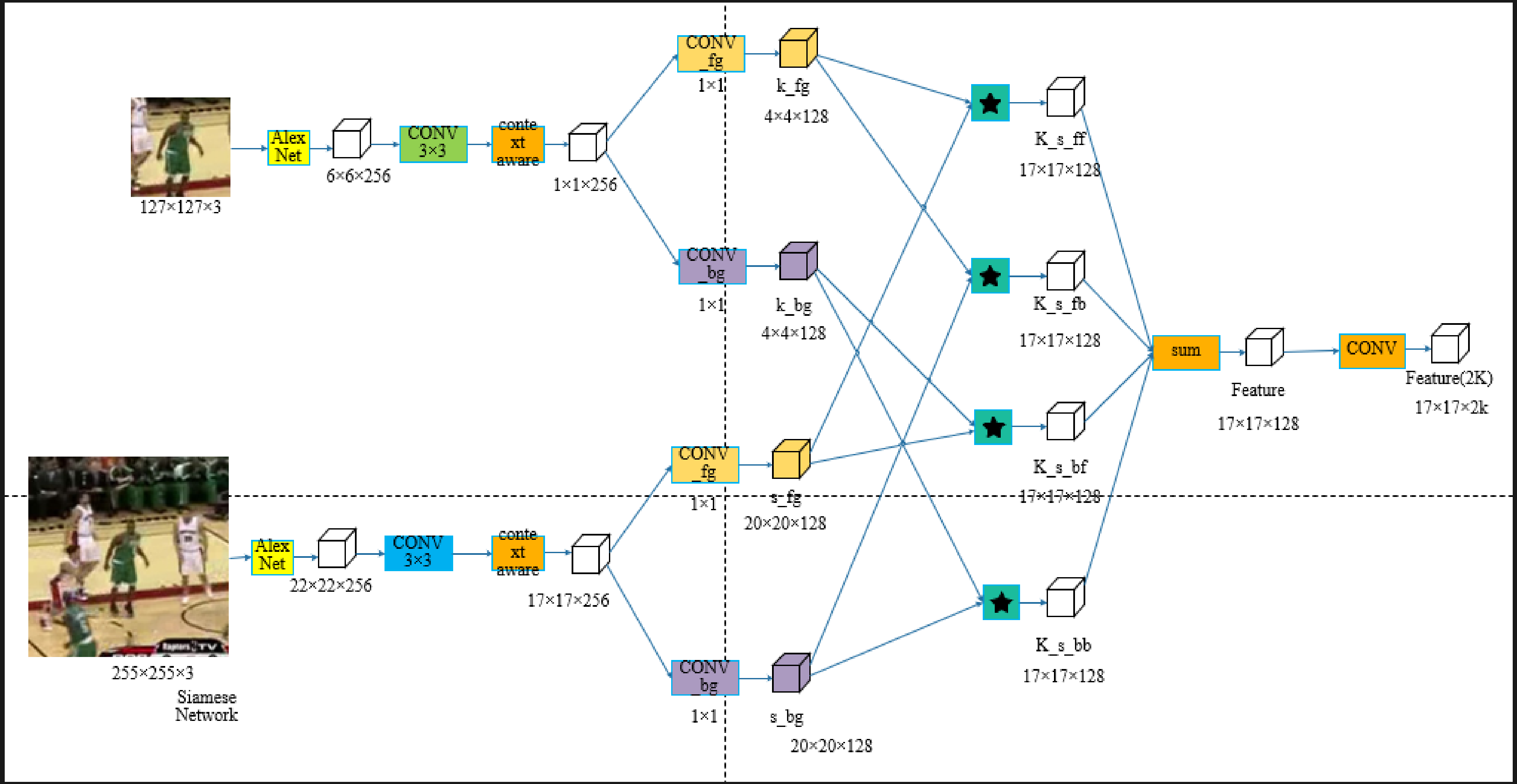


R:dist B:iou

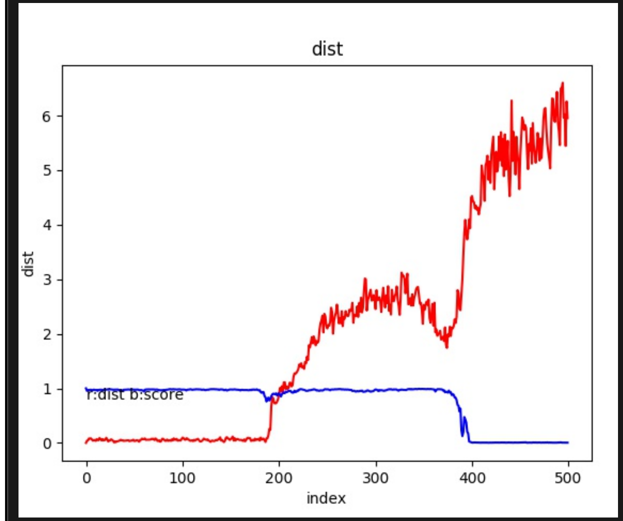


R:iou B:score

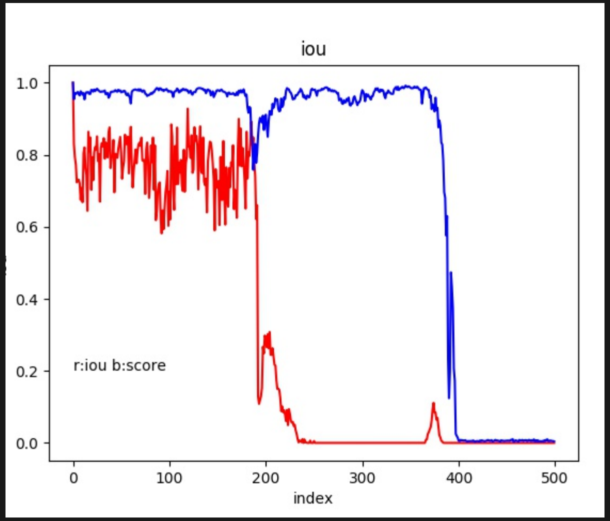
Disentangle & commonsense model



Disentangle & commonsense model

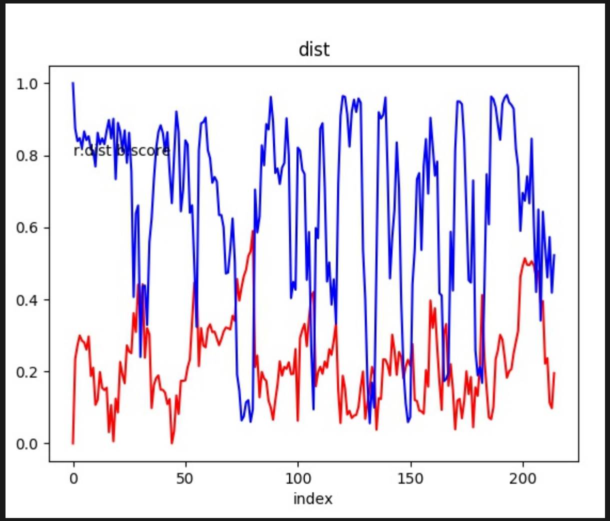


R:dist B:score

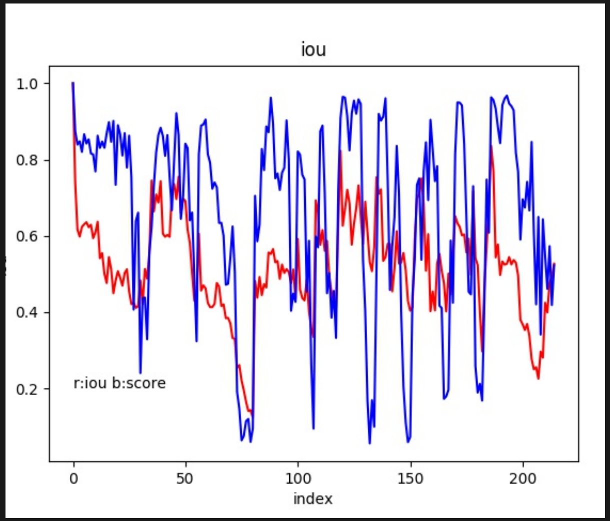


R:iou B:score

Disentangle & commonsense model

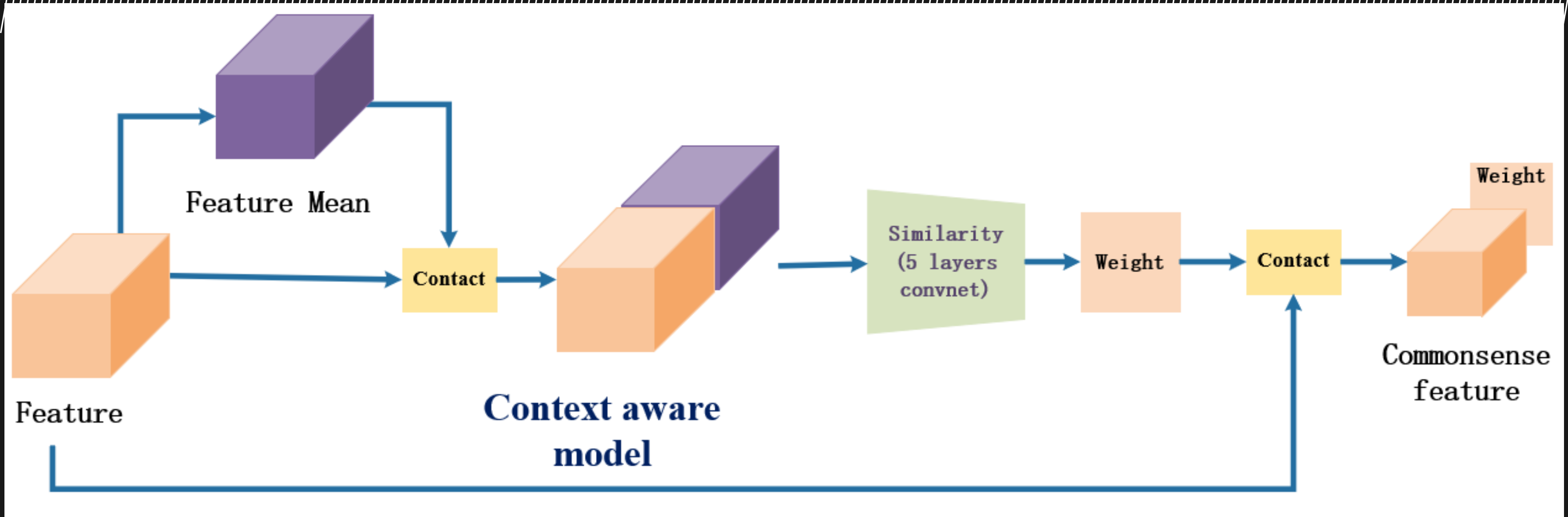


R:dist B:score



R:iou B:score

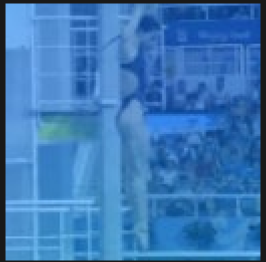
Commonsense feature



Kernel-weight

Kernel-feature

Kernel-feature (commonsense)



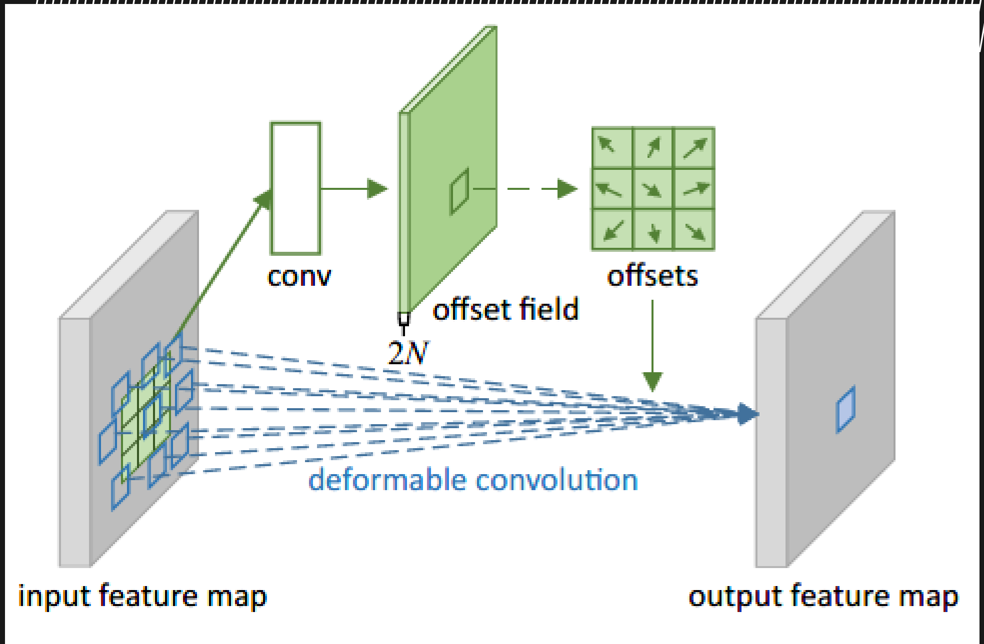
Search-feature

Search-feature(commonsense)

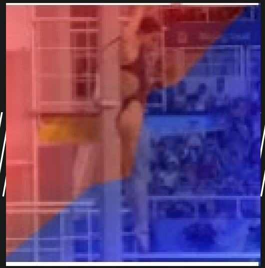
Search-weight

You See Future

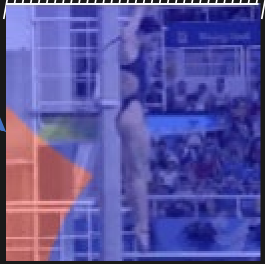
Deformable convolution



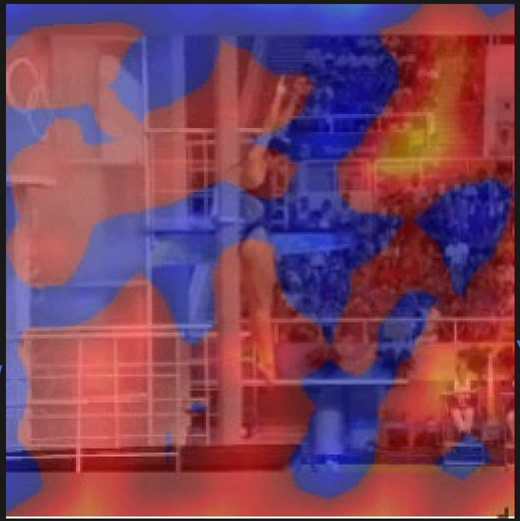
kernel-feature



kernel-offset



Kernel-feature(dcn)



search-offset

search-feature

search-feature
(dcn)

OTB 100

VOT

Deformable convolution

Tracker name	Success	Norm Precision	Precision
dcn	0.389	0.000	0.518

Tracker Name	Accuracy	Robustness	Lost Number	EAO
dcn	0.540	1.692	363.0	0.105

commonsense

Tracker name	Success	Norm Precision	Precision
commonsense_alex	0.399	0.000	0.536

Tracker Name	Accuracy	Robustness	Lost Number	EAO
commonsense_alex	0.549	1.659	356.0	0.106

Train rpn by pretrained model

Tracker name	Success	Norm Precision	Precision
train_rpn	0.377	0.000	0.500

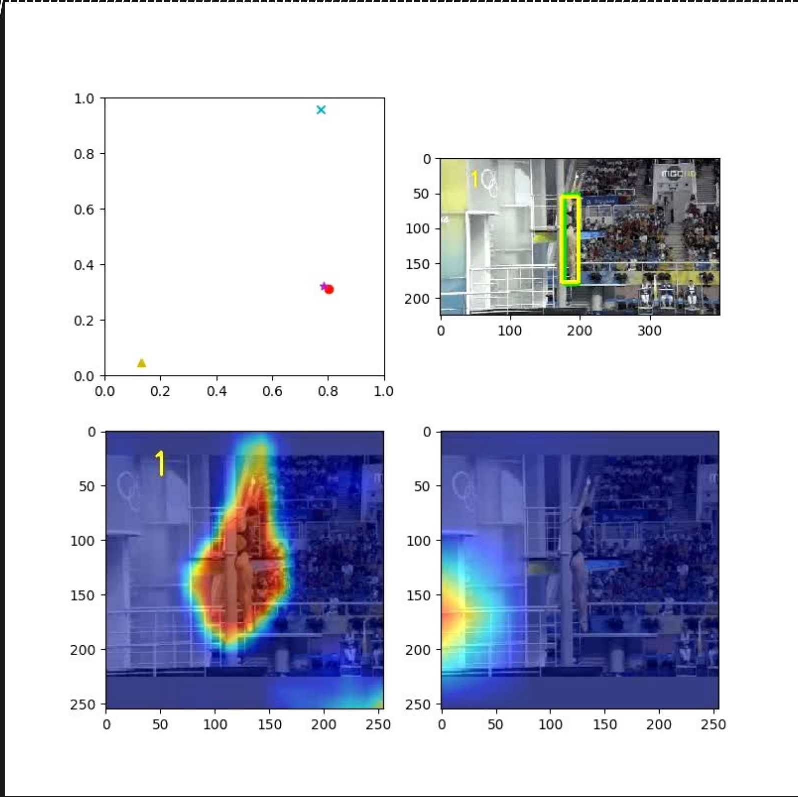
Tracker Name	Accuracy	Robustness	Lost Number	EAO
trainrpn	0.533	1.664	357.0	0.104

pretrained model

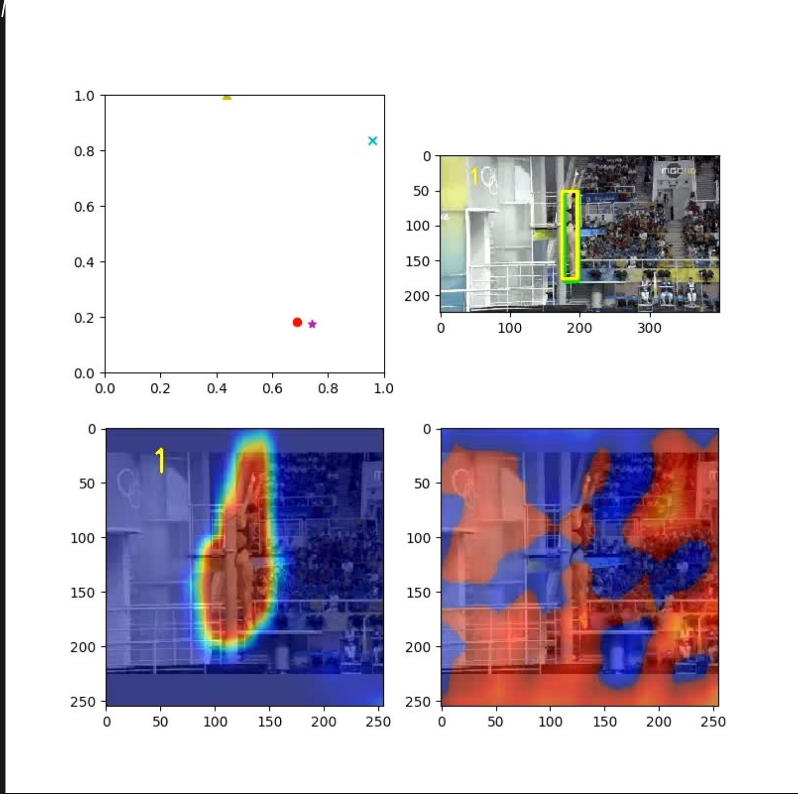
Tracker name	Success	Norm Precision	Precision
pretrain_rpn	0.416	0.000	0.555

Tracker Name	Accuracy	Robustness	Lost Number	EAO
pretrain_rpn	0.599	1.645	353.0	0.116

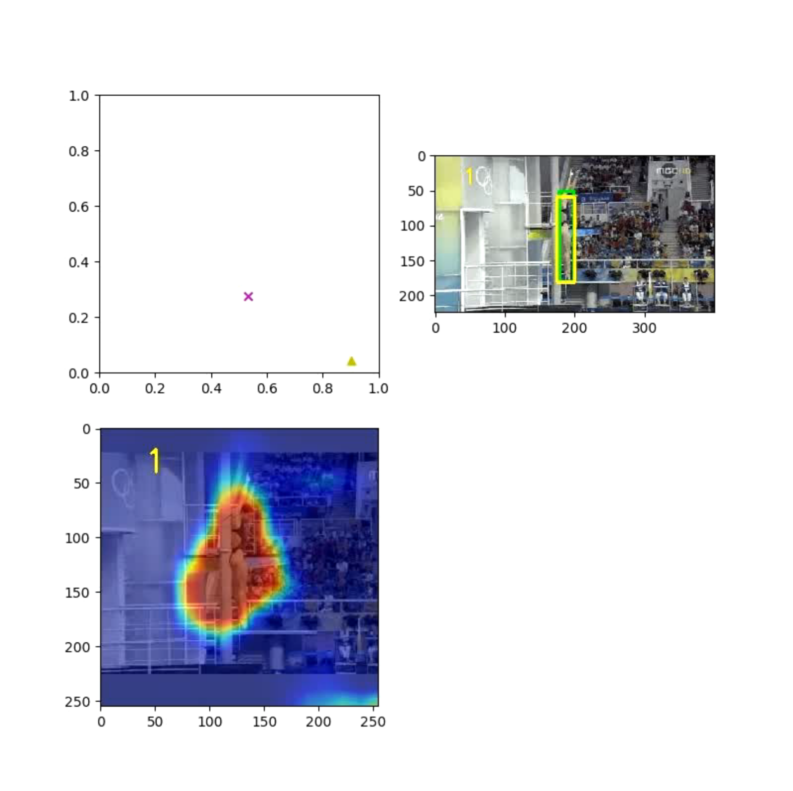
Commonsense



Deformable



RPN



Left-top: visualize features by tsne ----purple *: kernel-feature ; red o: (commonsense/dcn)kernel-feature ;
yellow ^:search-feature ; blue x: (commonsense/dcn)search-feature

Left-bottom: heatmap

right-top: tracking result

Right-bottom: commonse weight / deformable offset

Disentangle

How to explicitly model the two clusters as fg/bg?

1, Assign different weights

Hyperparameter that bias toward foreground correlation(is not the key-point)

2, Add an auxiliary loss

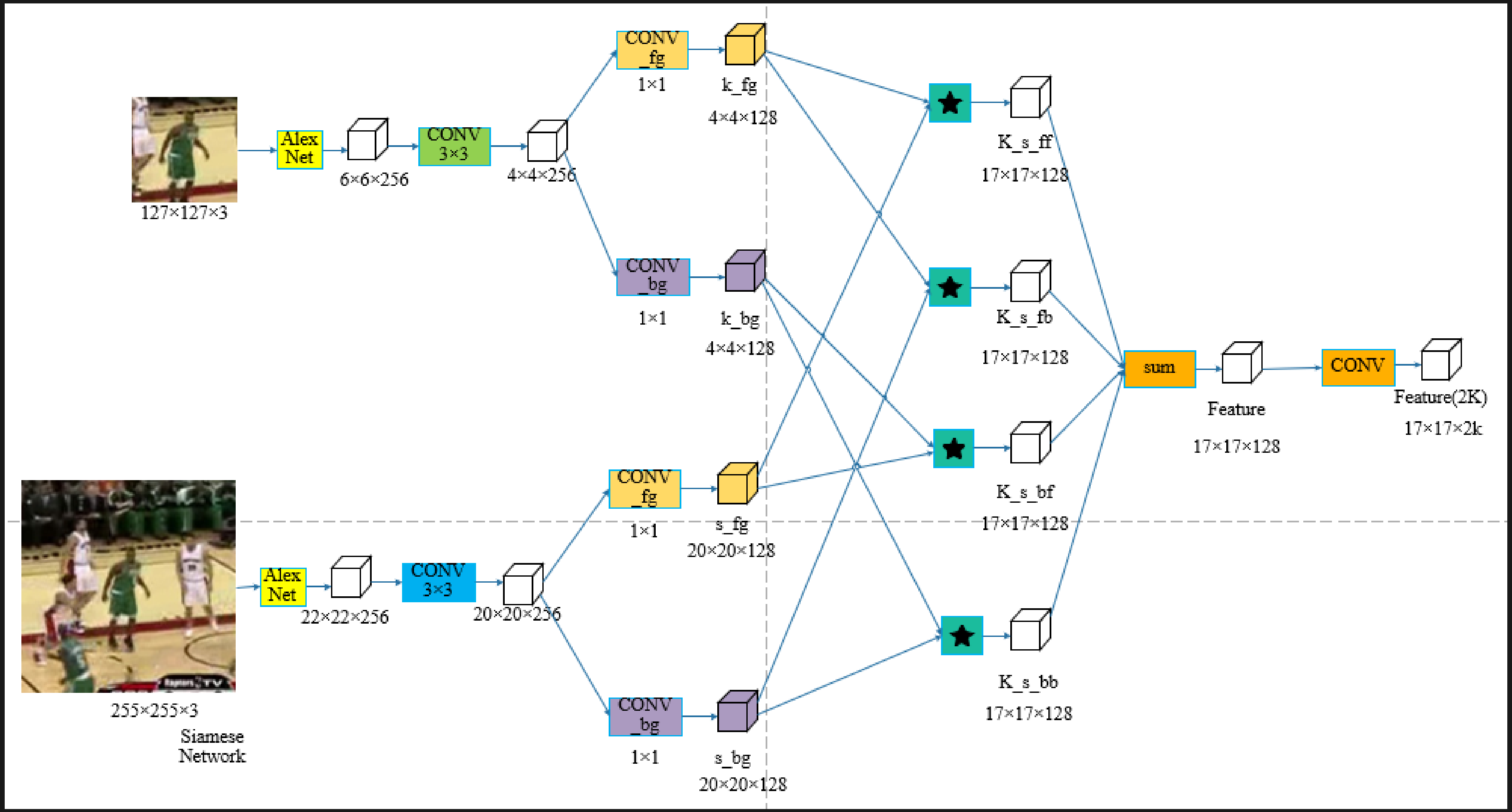
zero dot product between the fg/bg feature(can't ensure where is the foreground and where is the background)

3, reconstruction model

4, a new dataset

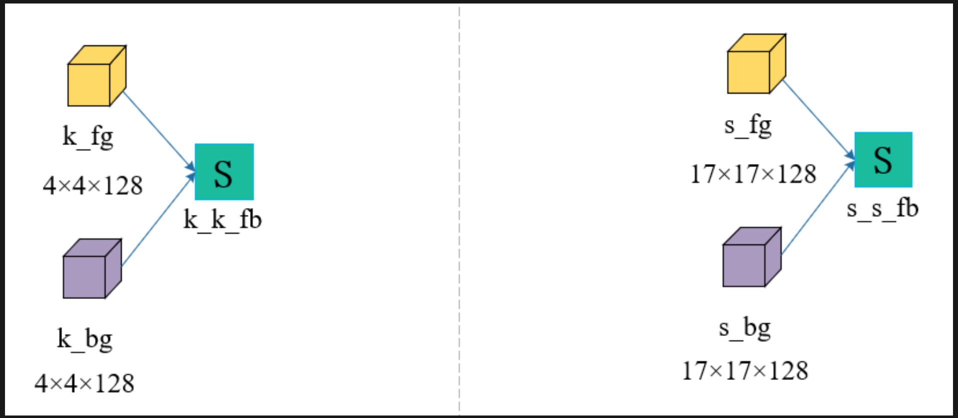
the background of datasets is random

Disentangle model

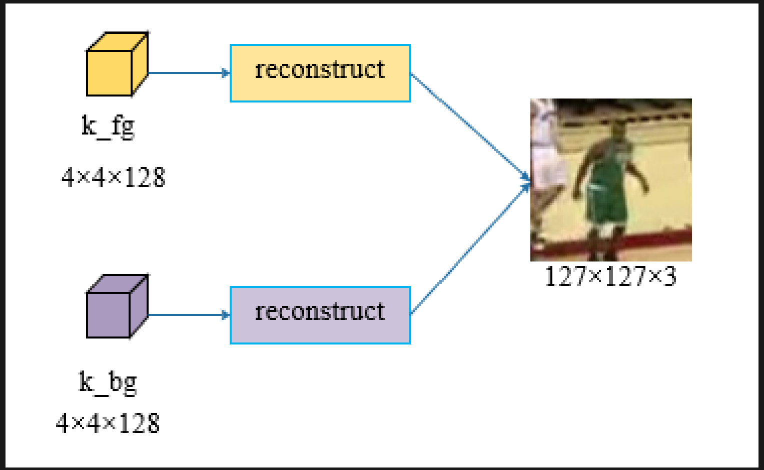


First , the extracted features are split into foreground features and background features to explore the relationship of them.

Disentangle + reconstruction model



Separate background and foreground at the feature level



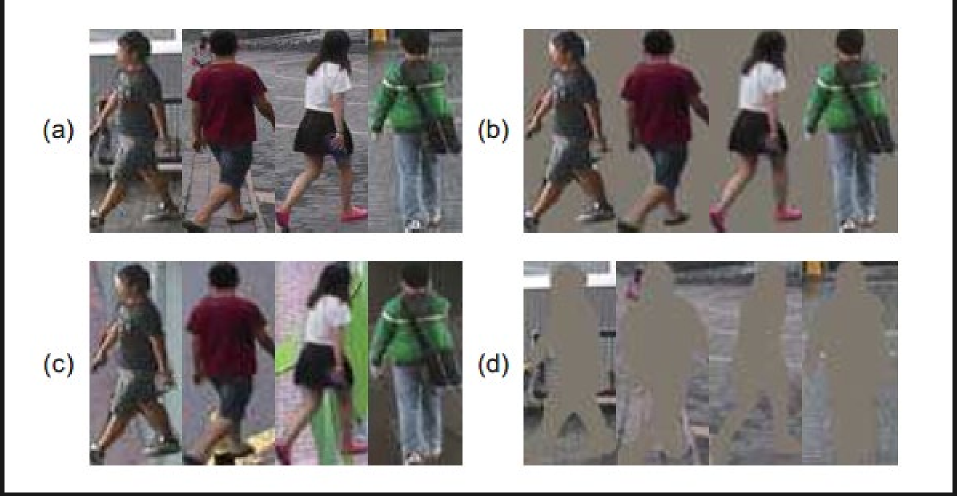
Problem : Ensure the separation of foreground and background by introducing similarity between foreground and background into the loss function. can't explicitly separate foreground and background.

method : In order to explicitly separate background and foreground, we design a reconstruction model, Refactoring features back to the original image, ensure the k_fg learned the feature of foreground.

Training dataset

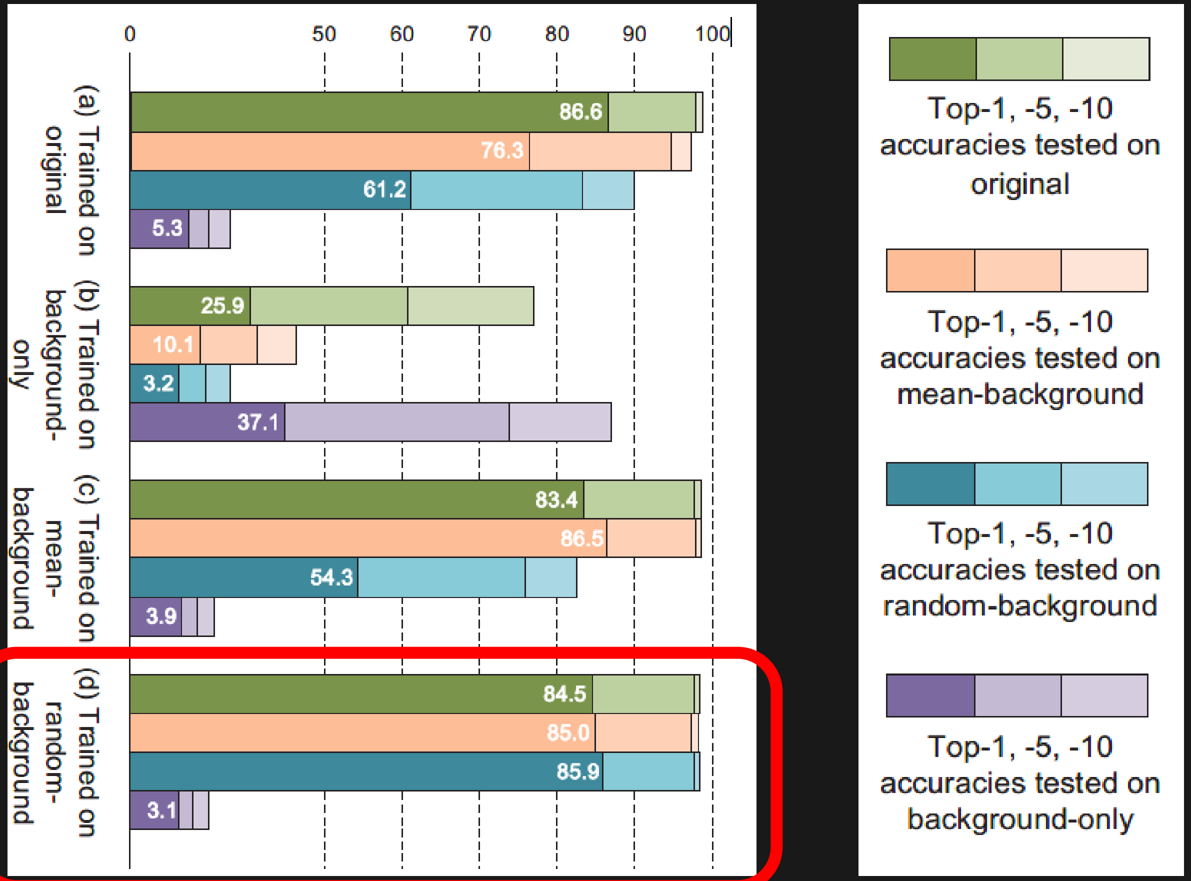
From a ReID task : Eliminating Background-bias for Robust Person Re-identification

Original-img Mean-background img



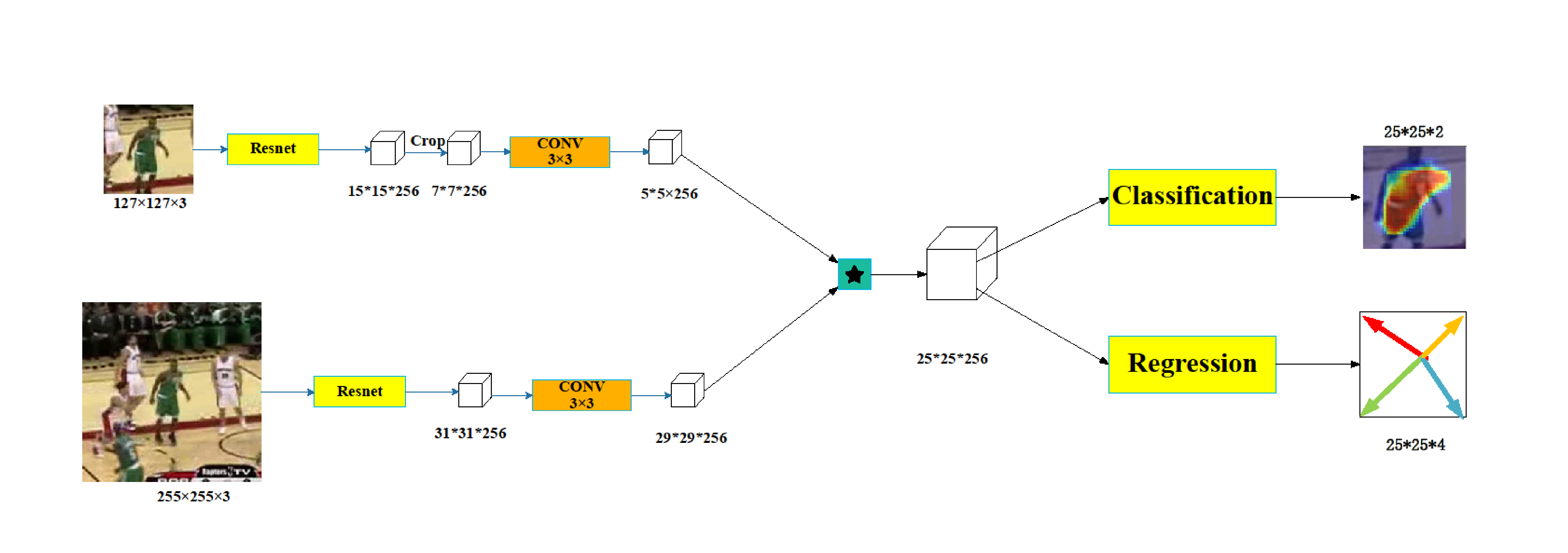
Random-background img Background-only img

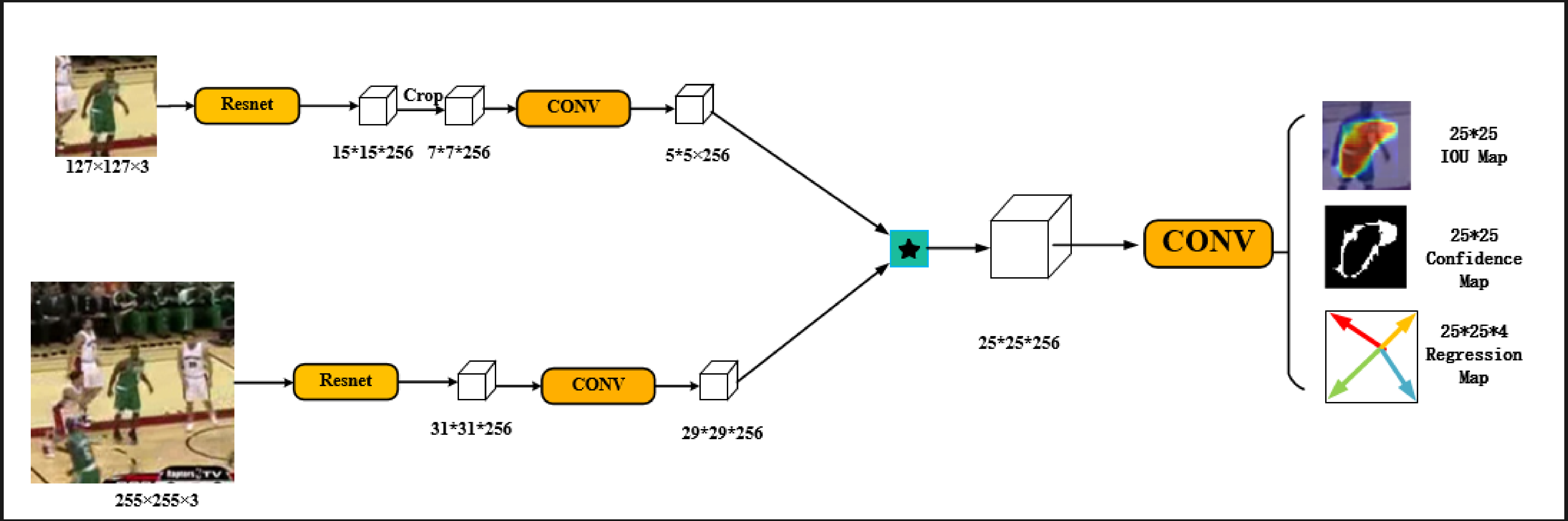
Experiment result:



The result of experiment which is trained on random-background is the best

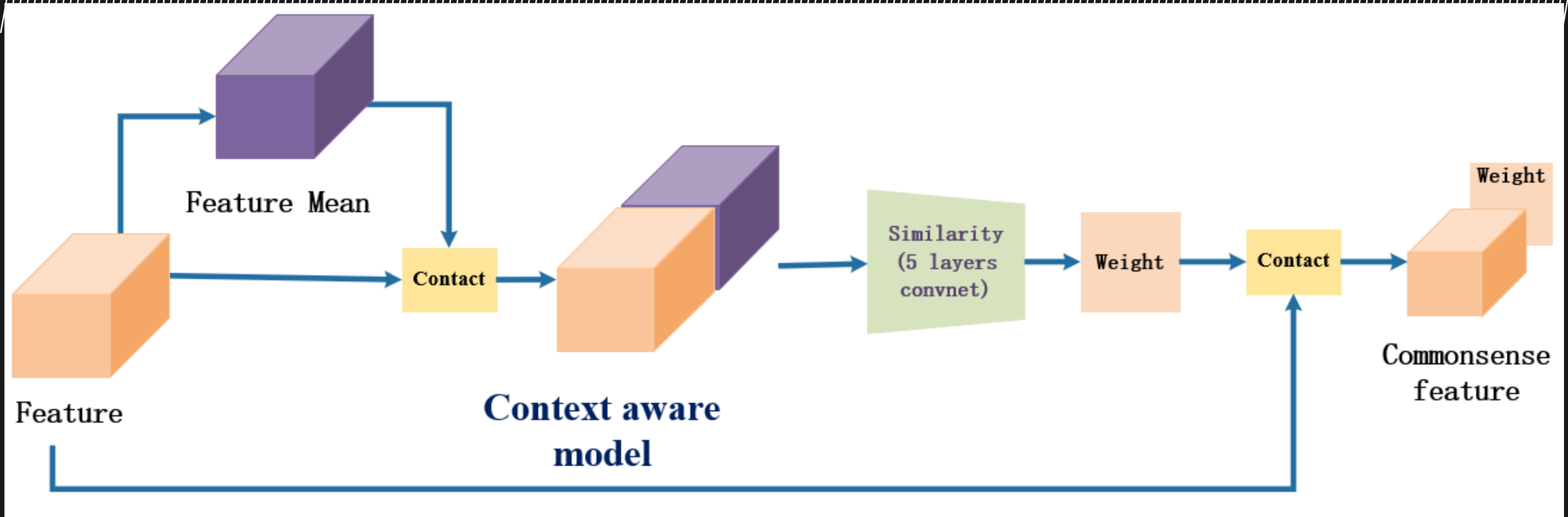
SOT New BaseLine





127×127×3
Resnet
15*15*256
Crop
7*7*256
CONV
5*5×256
255×255×3
Resnet
31*31*256
CONV
29*29*256
25*25*256
CONV
25*25
IOU Map
25*25
Confidence
Map
25*25*4
Regression
Map

Commonsense feature



Learn commonsense weight, directly contact feature and weight

OTB 100

Tracker name	Success	Norm Precision	Precision
commonsense_alex	0.399	0.000	0.536

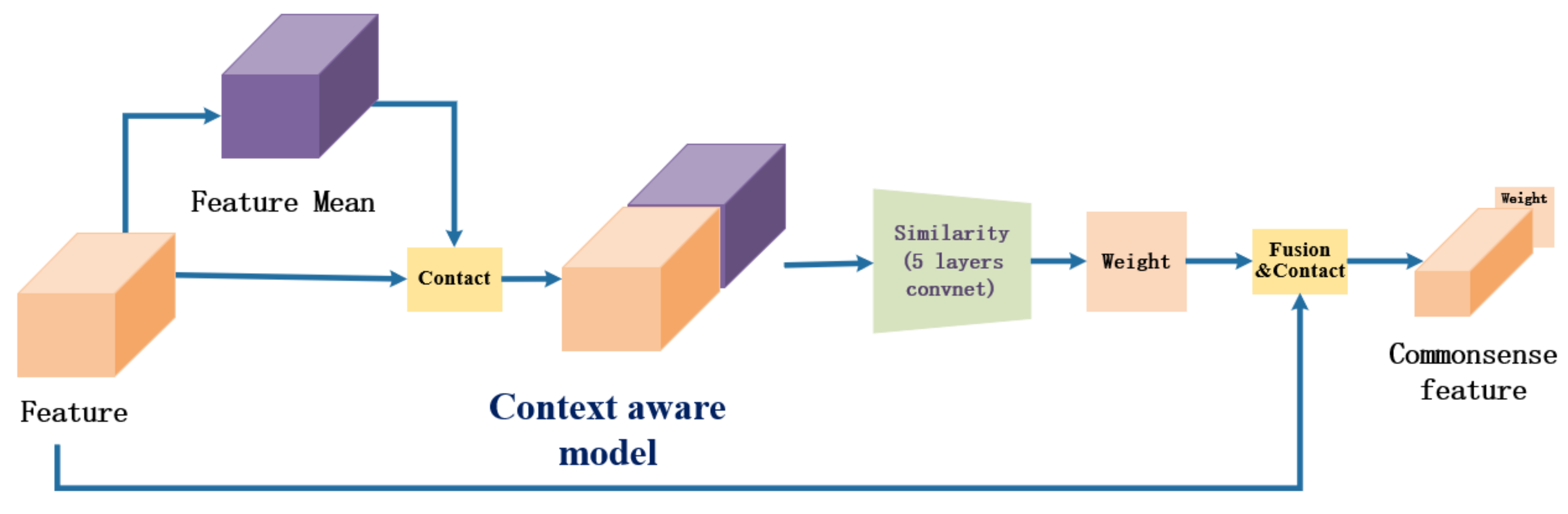
Tracker name	Success	Norm Precision	Precision
pretrain_rpn	0.416	0.000	0.555

VOT

Tracker Name	Accuracy	Robustness	Lost Number	EAO
commonsense_alex	0.549	1.659	356.0	0.106

Tracker Name	Accuracy	Robustness	Lost Number	EAO
pretrain_rpn	0.599	1.645	353.0	0.116

Commonsense feature



Learn commonsense weight, fusion weight and feature, then contact with weight

OTB 100

Tracker name	Success	Norm Precision	Precision
commonsense1_alex	0.385	0.000	0.512

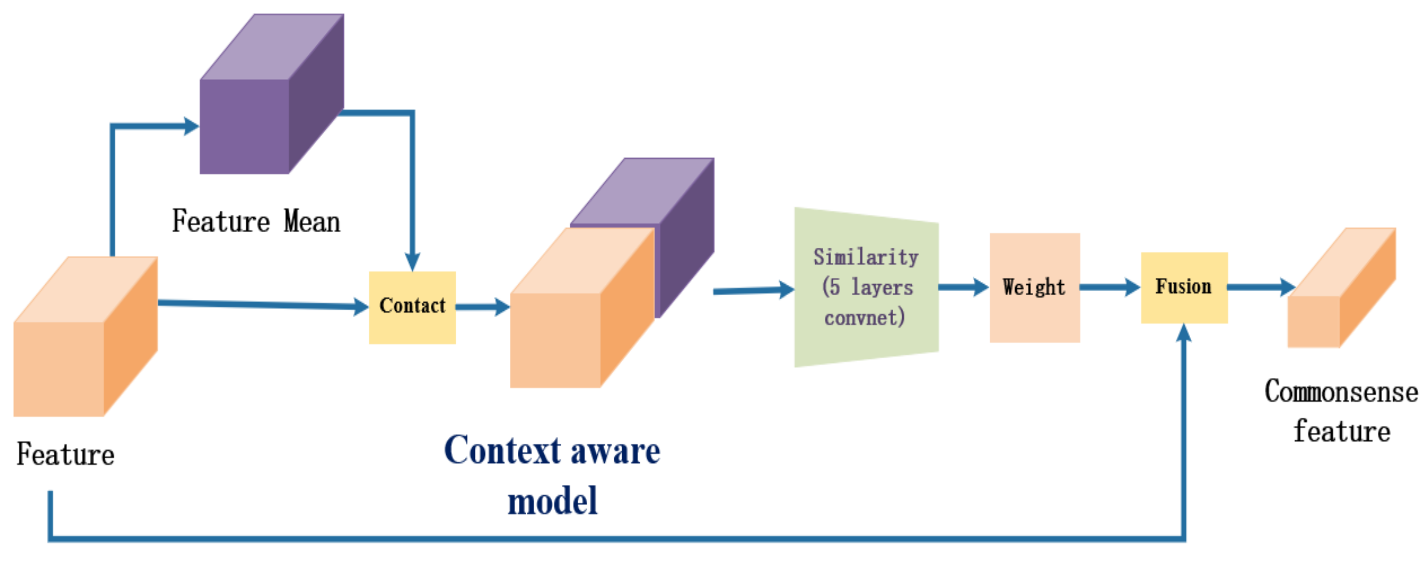
Tracker name	Success	Norm Precision	Precision
pretrain_rpn	0.416	0.000	0.555

VOT

Tracker Name	Accuracy	Robustness	Lost Number	EAO
commonsense1_alex	0.544	1.678	360.0	0.105

Tracker Name	Accuracy	Robustness	Lost Number	EAO
pretrain_rpn	0.599	1.645	353.0	0.116

Commonsense feature



Learn commonsense weight, fusion weight and feature

OTB 100

Tracker name	Success	Norm Precision	Precision
commonsense2_alex	0.383	0.000	0.510

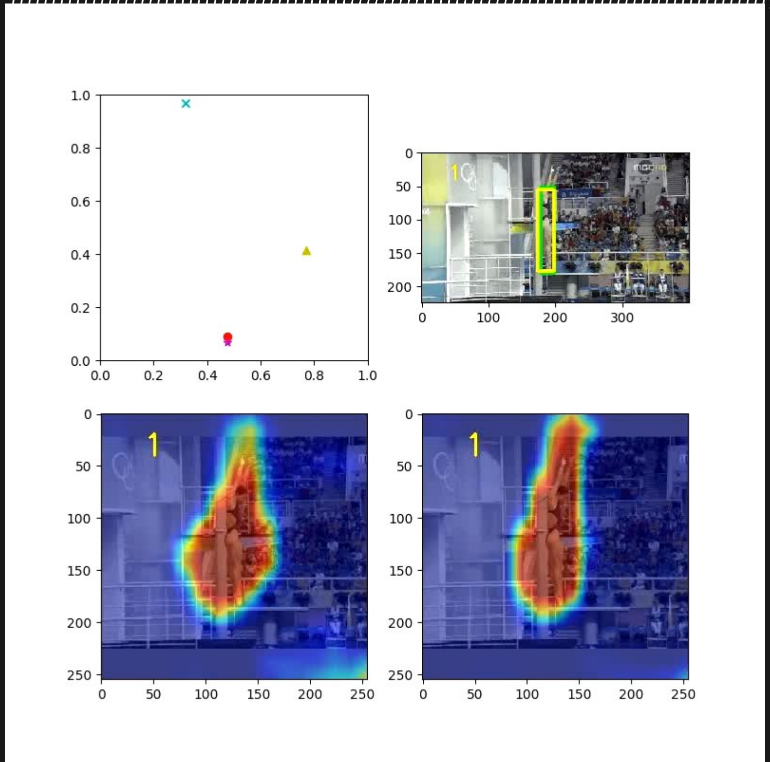
Tracker name	Success	Norm Precision	Precision
pretrain_rpn	0.416	0.000	0.555

VOT

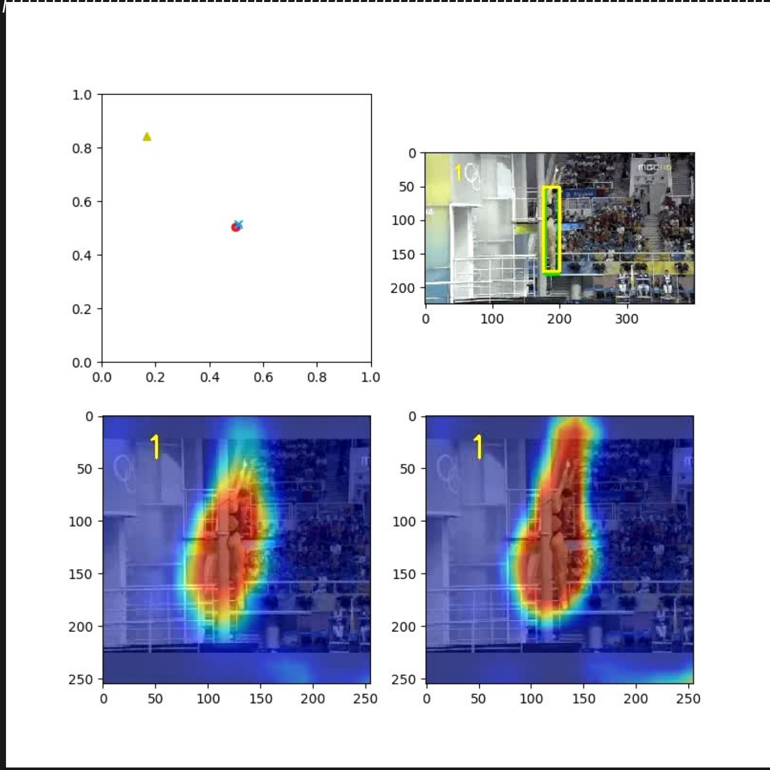
Tracker Name	Accuracy	Robustness	Lost Number	EAO
commonsense2_alex	0.535	1.725	370.0	0.105

Tracker Name	Accuracy	Robustness	Lost Number	EAO
pretrain_rpn	0.599	1.645	353.0	0.116

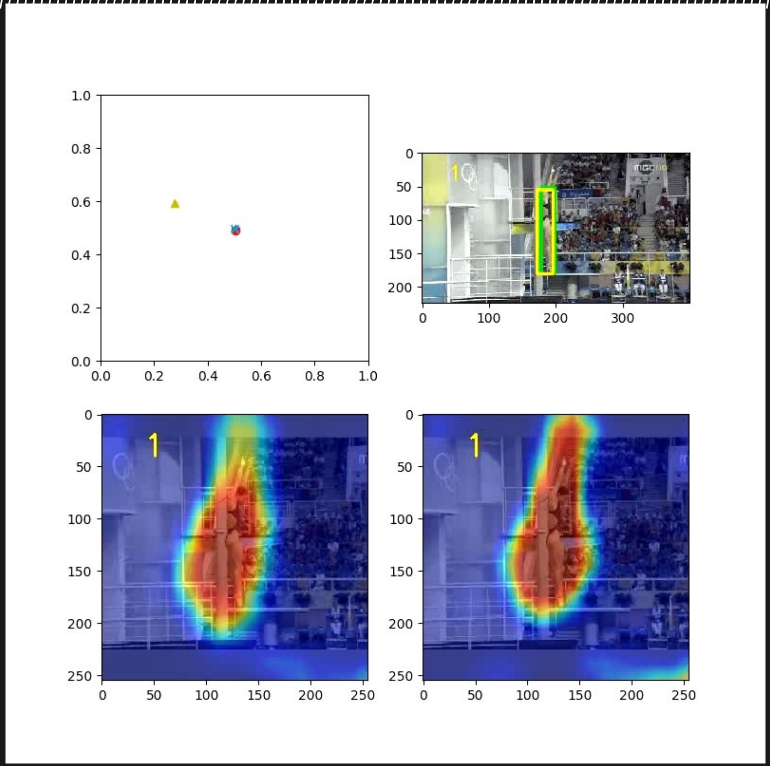
directly contact feature and weight



fusion weight and feature, then contact with weight



fusion weight and feature



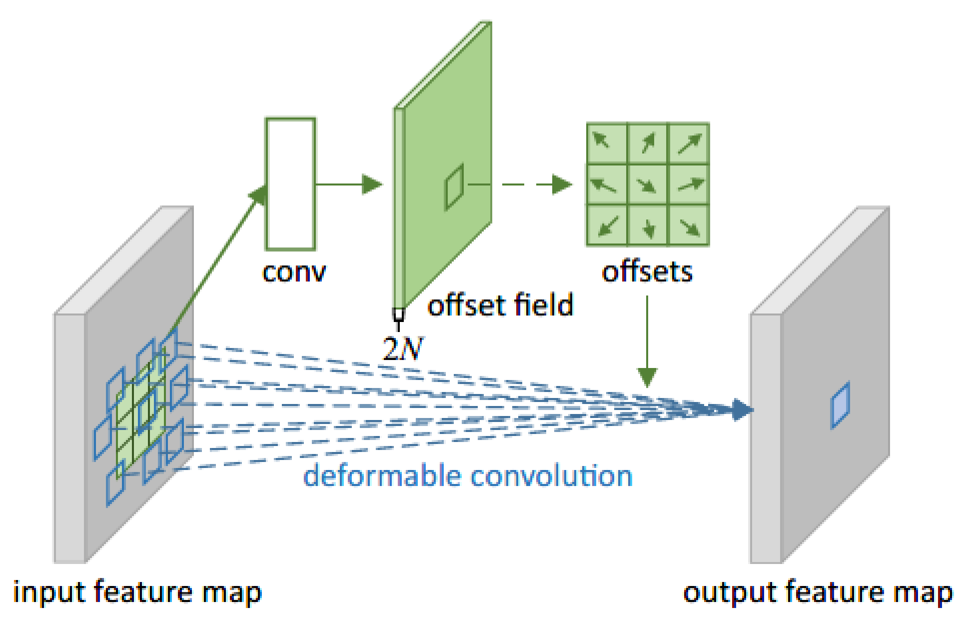
Left-top: visualize features by tsne ----purple *: kernel-feature ; red o: (commonsense)kernel-feature ;
yellow ^:search-feature ; blue x: (commonsense)search-feature

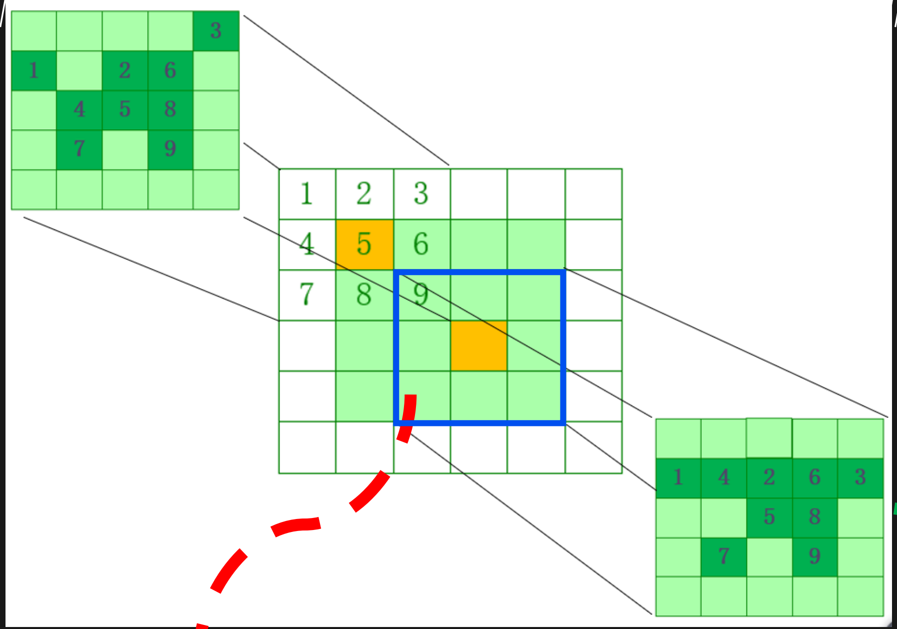
Left-bottom: heatmap
right-top: tracking result
Right-bottom: original-heatmap

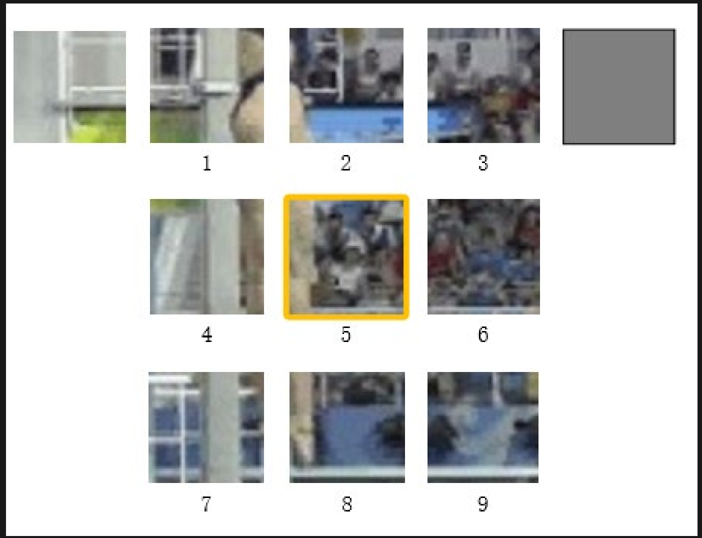
UISEE

You See Future

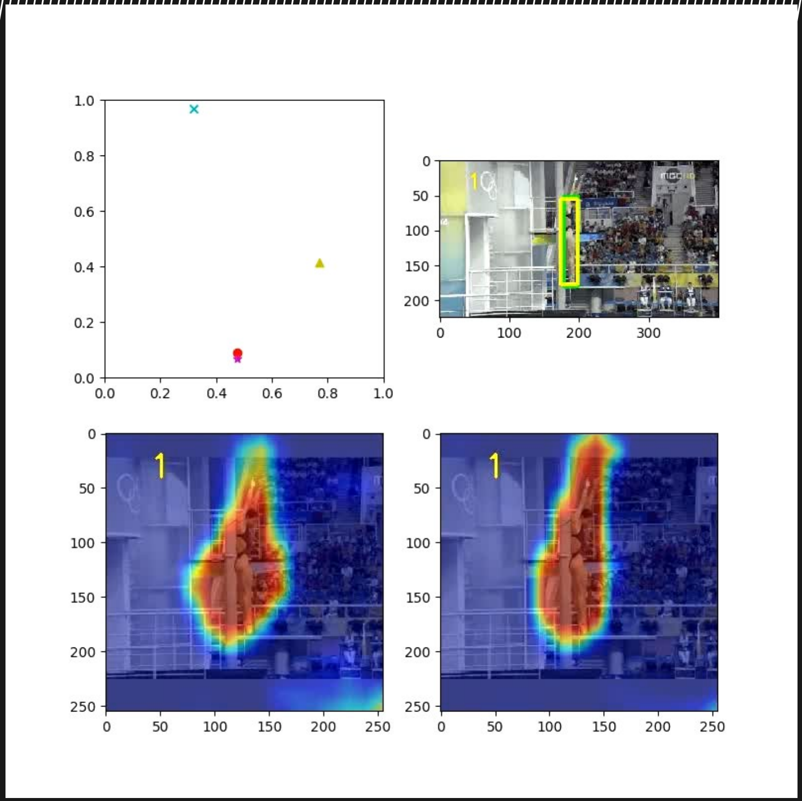
Deformable convolution



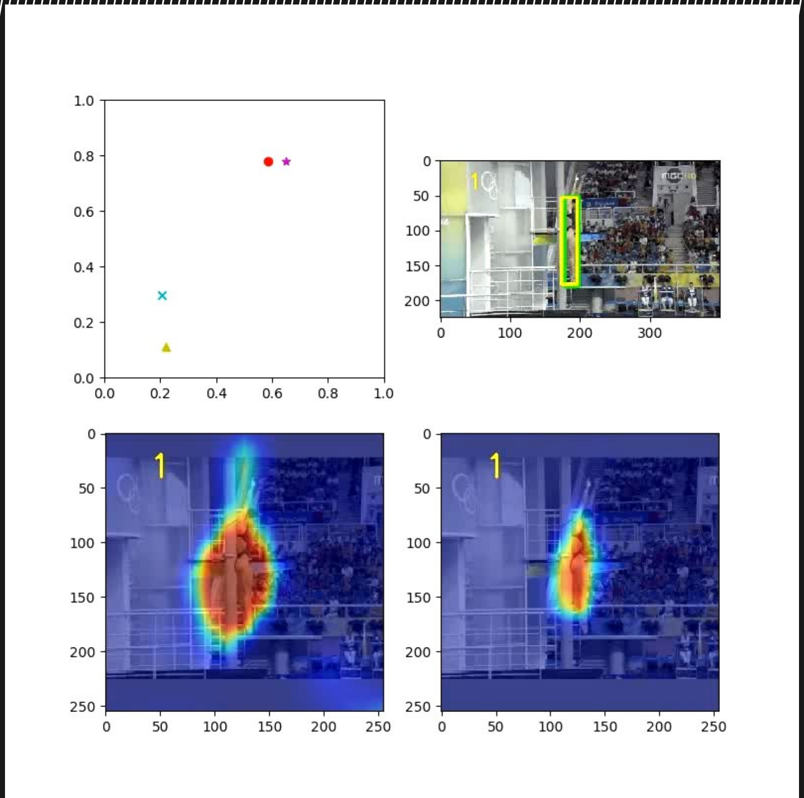




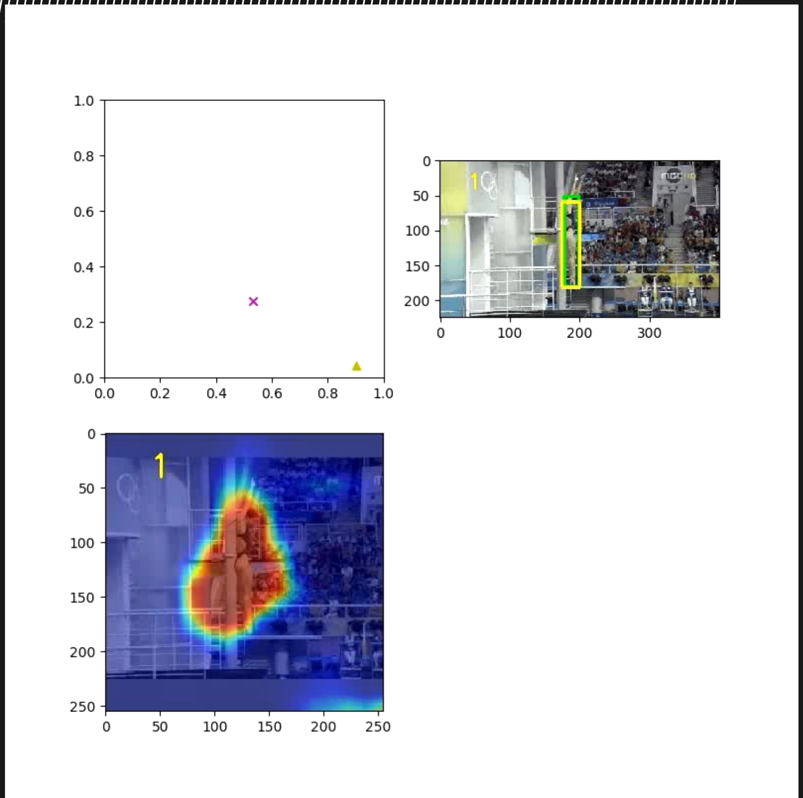
Commonsense



Deformable



SiamRPN



Left-top: visualize features by tsne ----purple *: kernel-feature ; red o: (commonsense/dcn)kernel-feature ;
yellow ^:search-feature ; blue x: (commonsense/dcn)search-feature

Left-bottom: heatmap

right-top: tracking result

Right-bottom: original-heatmap

OTB 100

VOT

Deformable convolution

Tracker name	Success	Norm Precision	Precision
dcn	0.389	0.000	0.518

Tracker Name	Accuracy	Robustness	Lost Number	EAO
dcn	0.540	1.692	363.0	0.105

commonsense

Tracker name	Success	Norm Precision	Precision
commonsense_alex	0.399	0.000	0.536

Tracker Name	Accuracy	Robustness	Lost Number	EAO
commonsense_alex	0.549	1.659	356.0	0.106

Train siamrpn by pretrained model

Tracker name	Success	Norm Precision	Precision
train_rpn	0.377	0.000	0.500

Tracker Name	Accuracy	Robustness	Lost Number	EAO
trainrpn	0.533	1.664	357.0	0.104

Siamrpn pretrained model

Tracker name	Success	Norm Precision	Precision
pretrain_rpn	0.416	0.000	0.555

Tracker Name	Accuracy	Robustness	Lost Number	EAO
pretrain_rpn	0.599	1.645	353.0	0.116

Disentanglement with Capsule

Have Done

1, Further training to improve the performance
2, Extend to more parts, body part disentangling

TODO

Try adding auxiliary loss or foreground mask as supervision

Reconstruct the learnt fg/bg feature to the template, use the reconstruction MSE loss as supervision

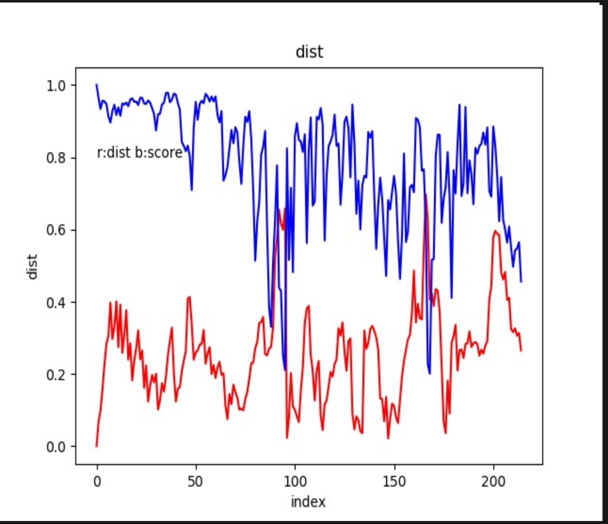
Multiple Part Disentangling

1, Extend the fg/bg model to more parts
2, Eliminate the weight hyperparameter
3, Network assigns the weight to correlation of each part, as is similar in SENet

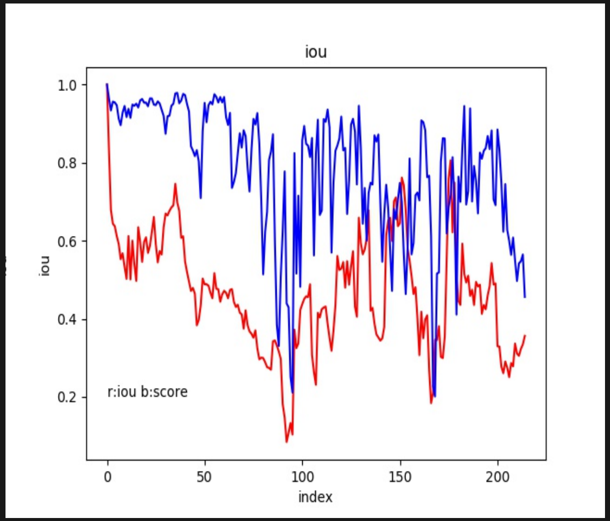
Results

Better than the fg/bg model but still well below the baseline …
Observation:
The learnt weights all converge to 0.5, might be the cause of poor performance

Multipart Disentangle model

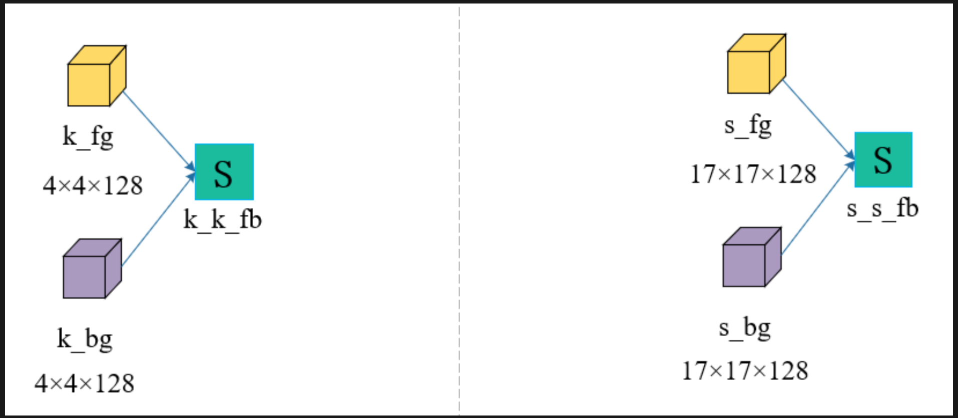


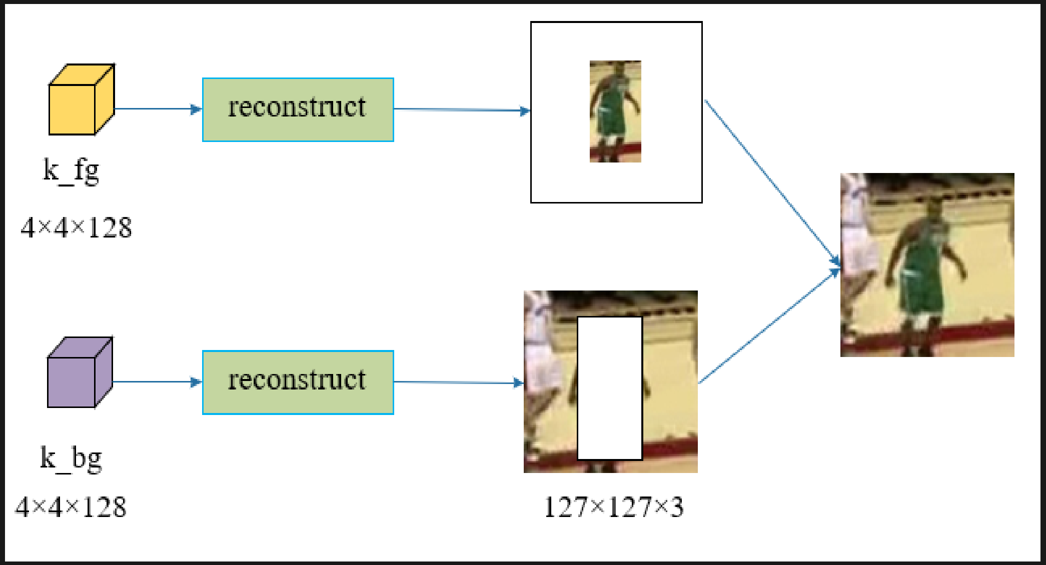
R:dist B:score



R:iou B:score

Disentangle + reconstruction model



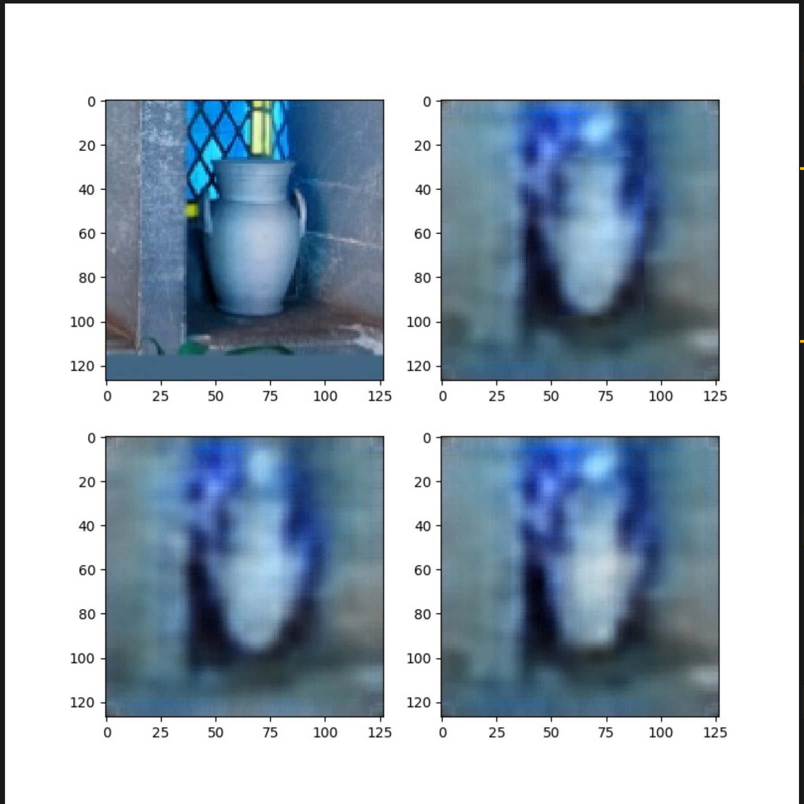


Separate background and foreground at the feature level

Problem : Ensure the separation of foreground and background by introducing similarity between foreground and background into the loss function. can't explicitly separate foreground and background.

method : In order to explicitly separate background and foreground, we design a reconstruction model, Refactoring features back to the original image, ensure the k_fg learned the feature of foreground.

Reconstruction



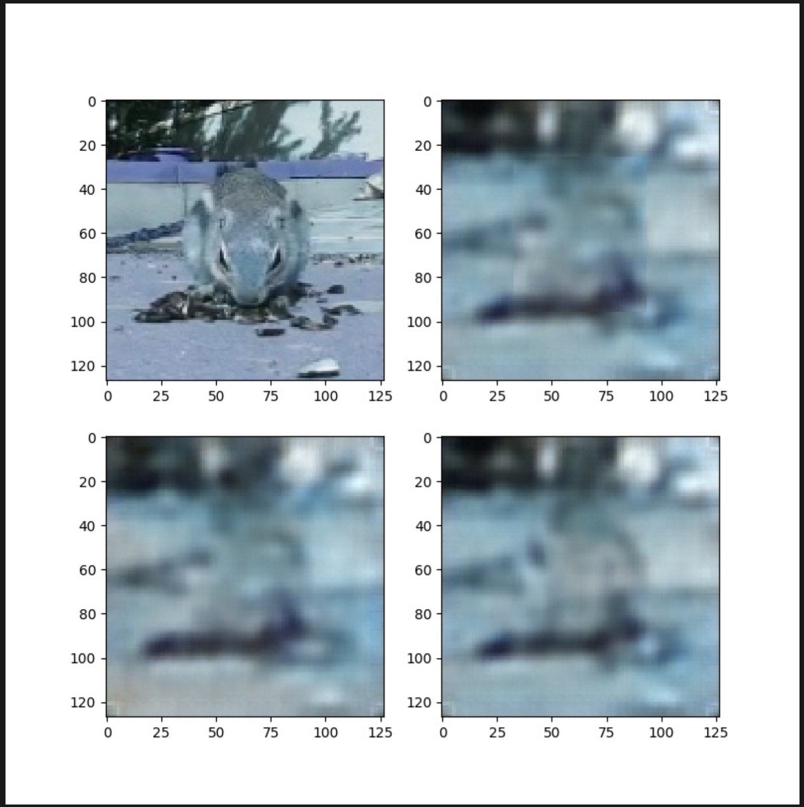
Reconstruction result：

Top-left:
Original image

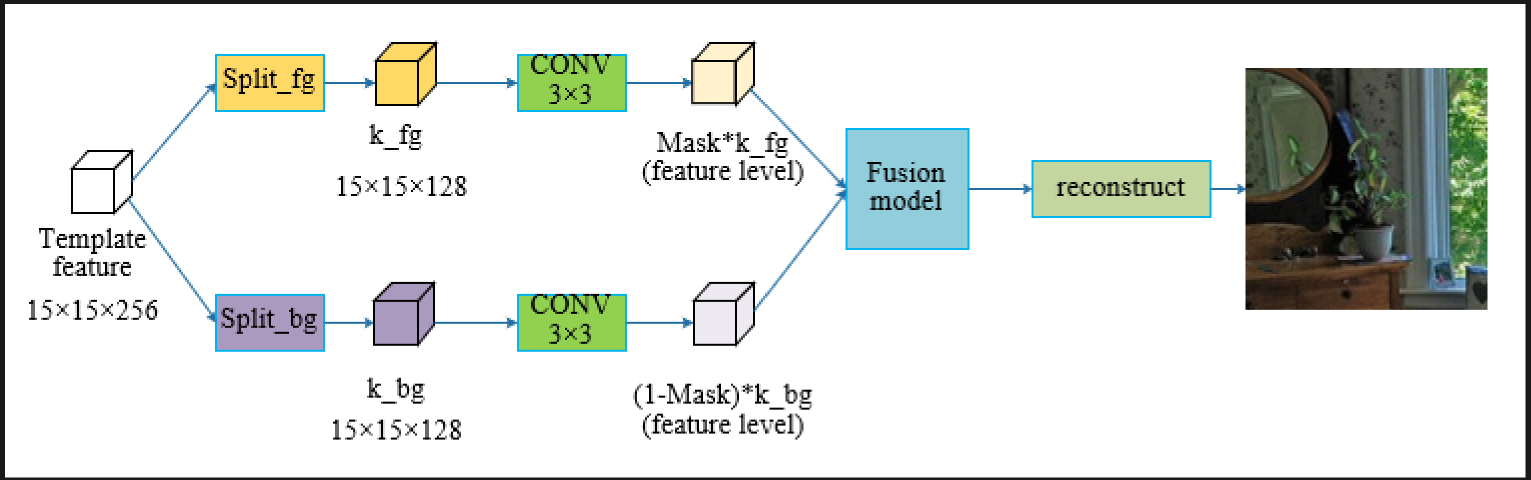
Top-right:
combine reconstructed image

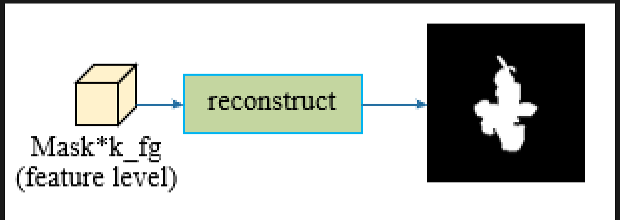
Bottom-left:
reconstruct foreground feature

Bottom-right:
reconstruct background feature



Disentangle and Reconstruction model details



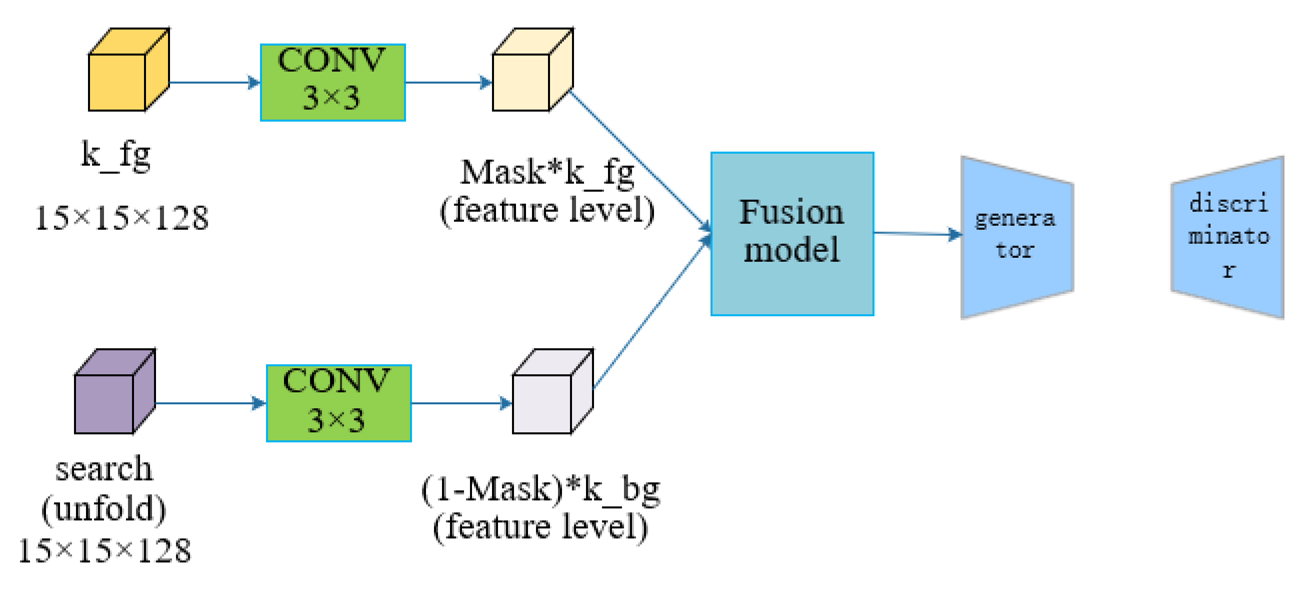


Train disentangle model and reconstruction model:

Separate the template feature into foreground and background by disentangle model, then learn a mask by a conv-net, fusion foreground feature and background feature to reconstruct image, compute the difference of original image and reconstructed image to train disentangle model and reconstruction model.

Reconstruct mask image by reconstruction model to make sure the mask is accurate enough.

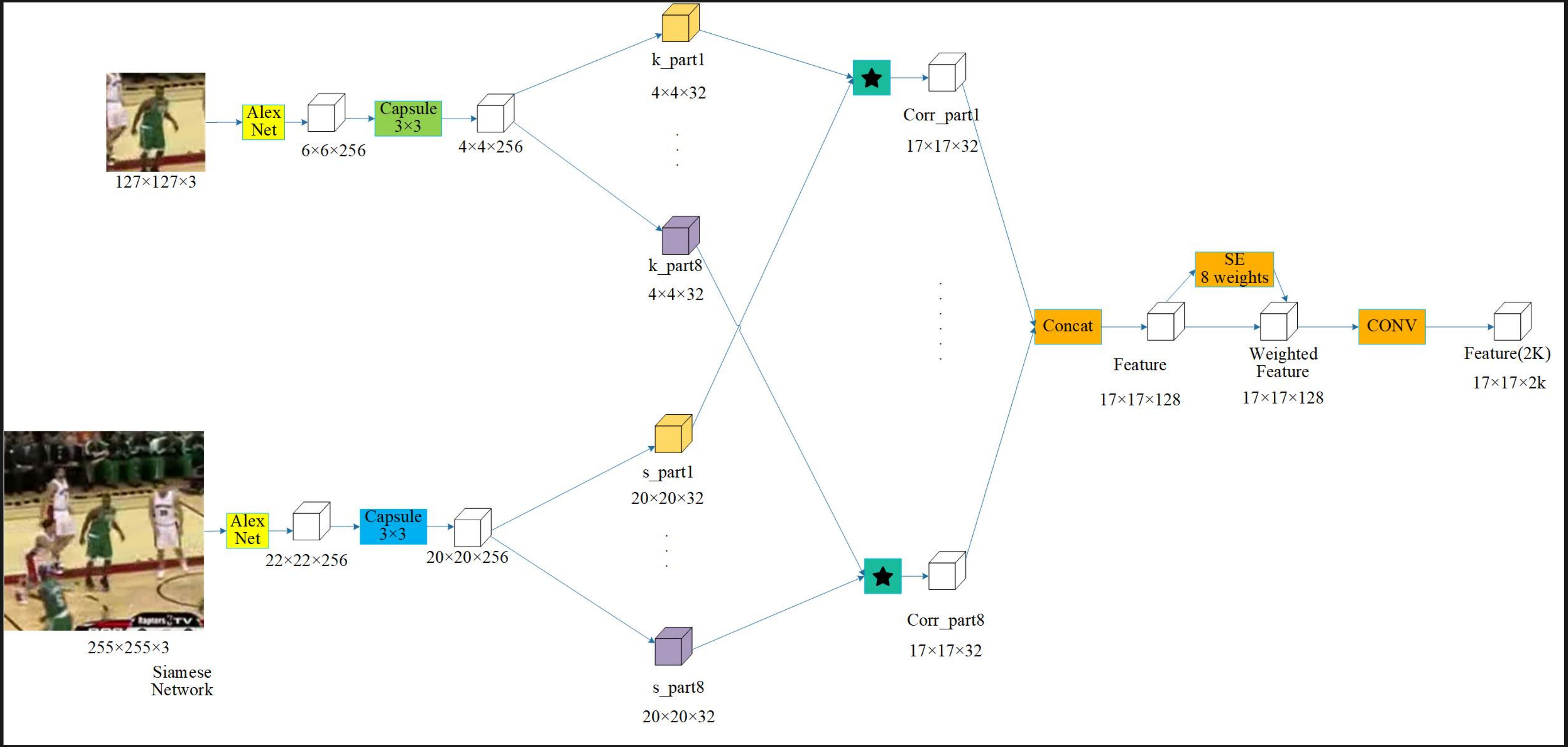
GAN model details



After completing the training of the disentangle and reconstruction models, we get the foreground feature of template and the background feature of search by using these two models, then combine these two features into the generator to generate image, using discriminator to discriminate if the reconstructed image and original image is the same as the result of classification.

if we can get a accurate mask feature and mask image, the result of reconstruct mask will instead regression branch.

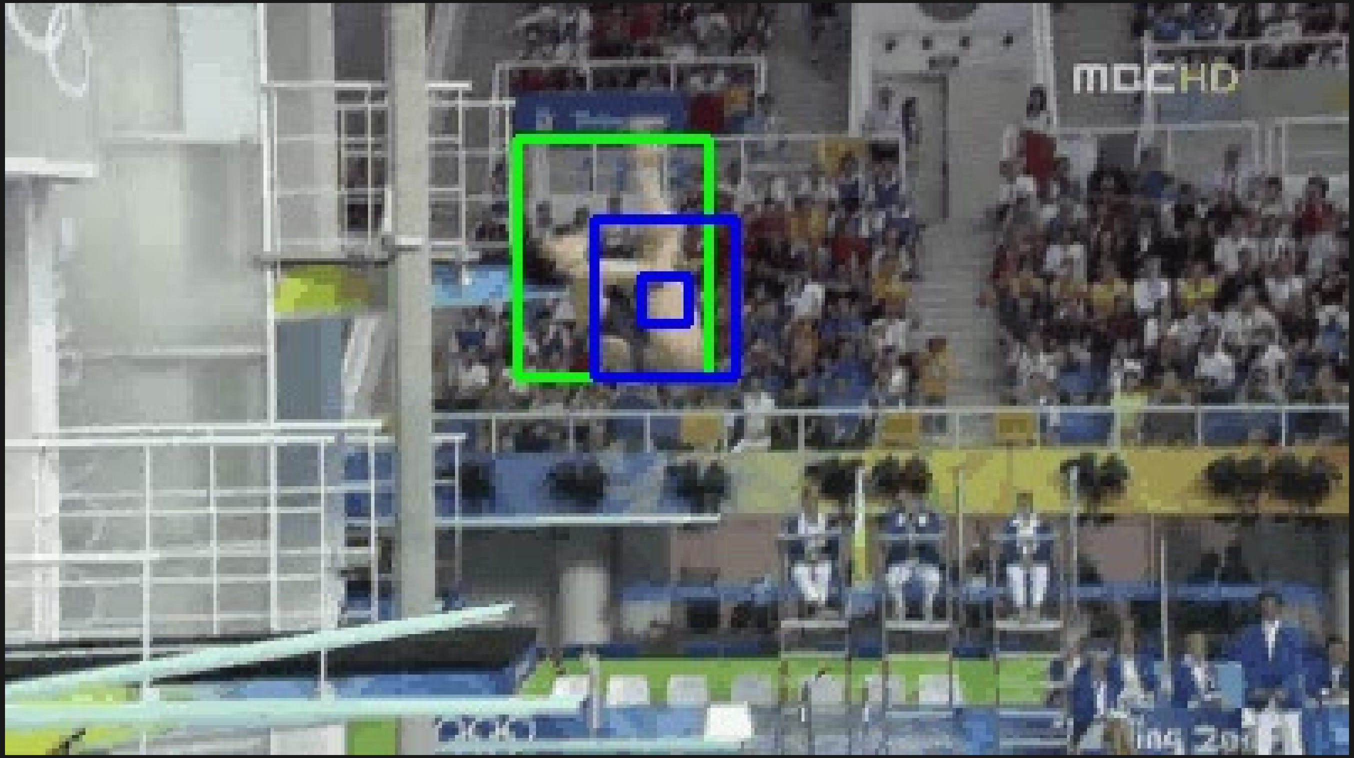
Disentanglement with Capsule



Multiple Part Disentangling

Problem

The regression branch doesn't work well in test phase.
The bounding box is almost anchor itself.
However, it works well on training and **validation**.



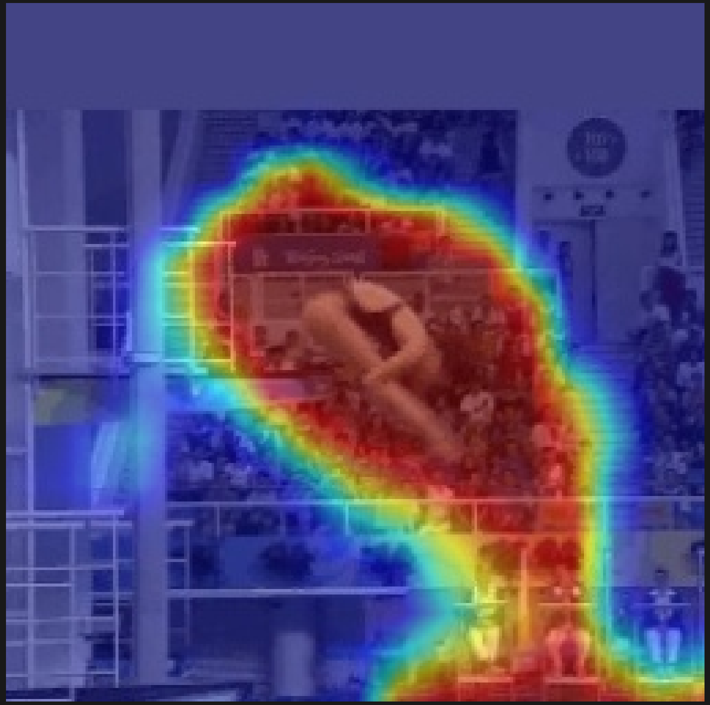
Multiple Part Disentangling

Guess: BN layer fails in small batch size configuration

Solution

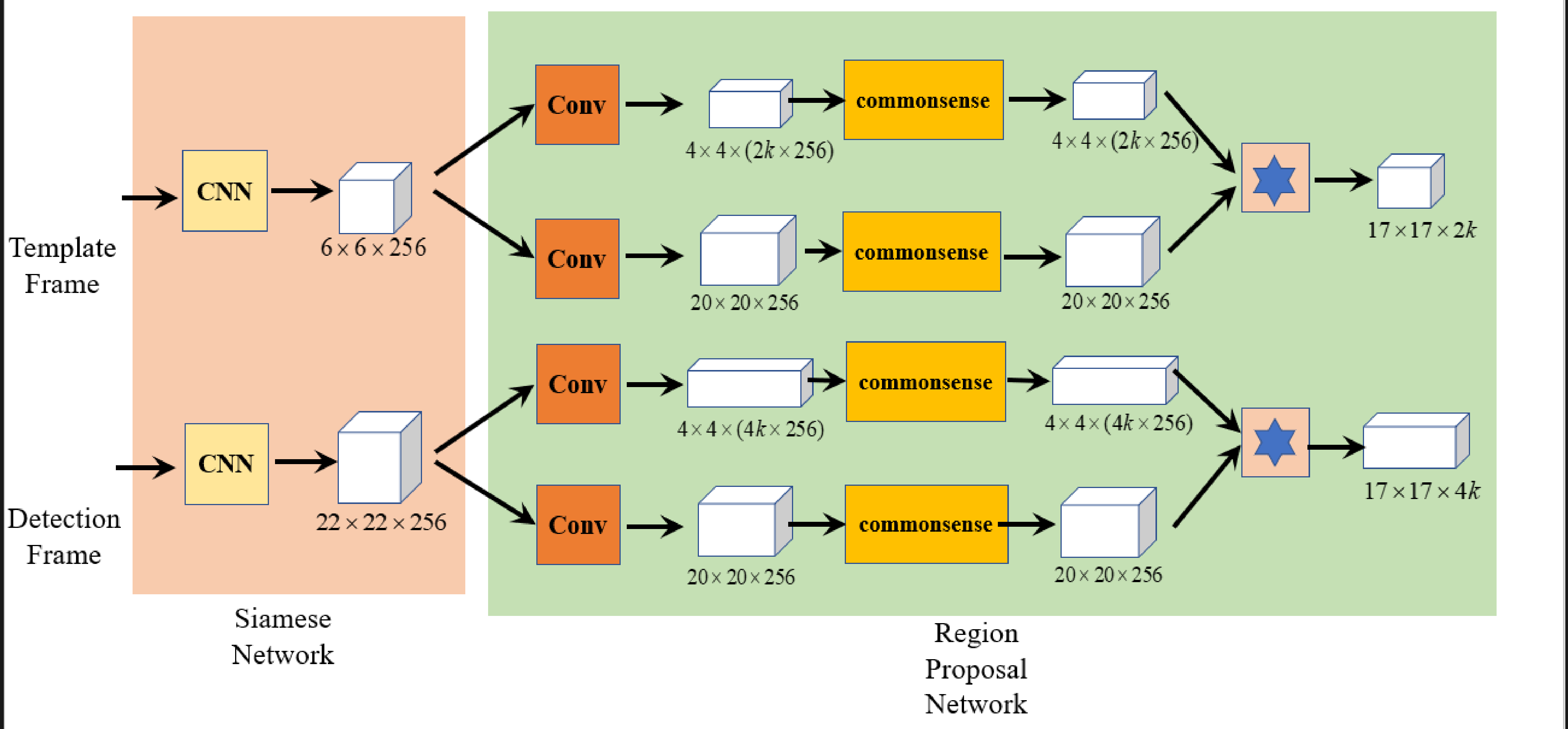
Replace the BN layer into batch size irrelevant GN layer

Other settings:
partition the input channel to more capsules (1x1 kernel)

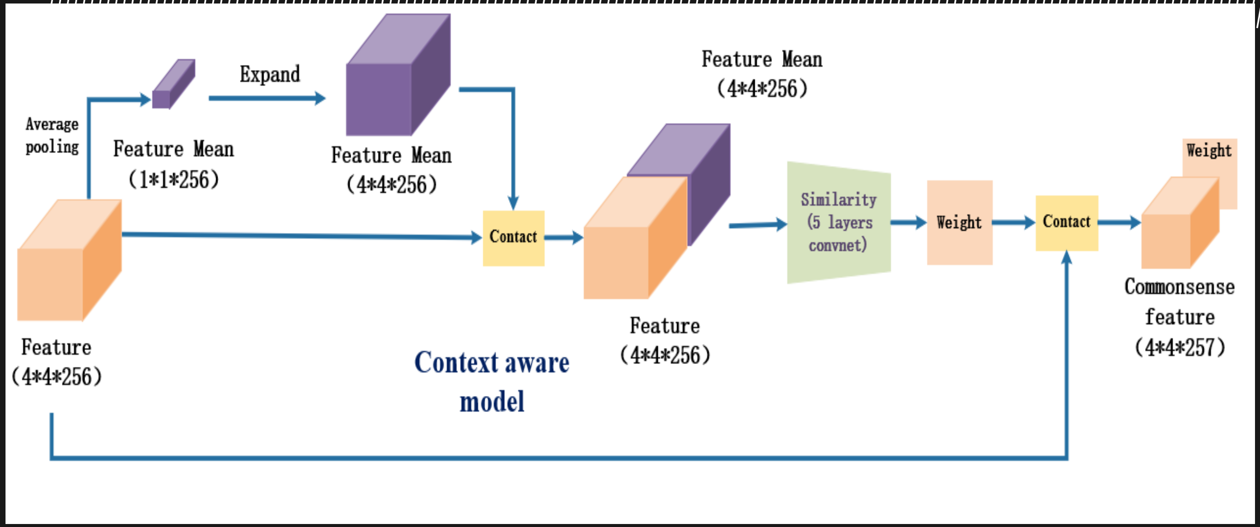


Results are same or worse…

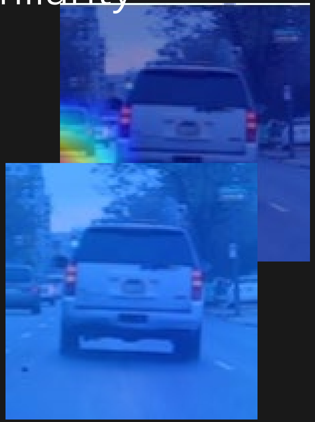
Siamese RPN with common sense



Commonsense feature

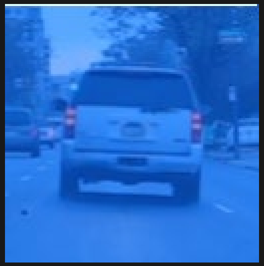


Template Similarity
\weight



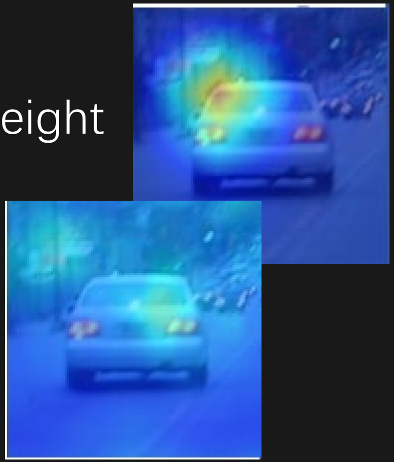
Search Original
feature

Template Commonsense
feature



Similarity weight

Original feature



Commonsense
feature

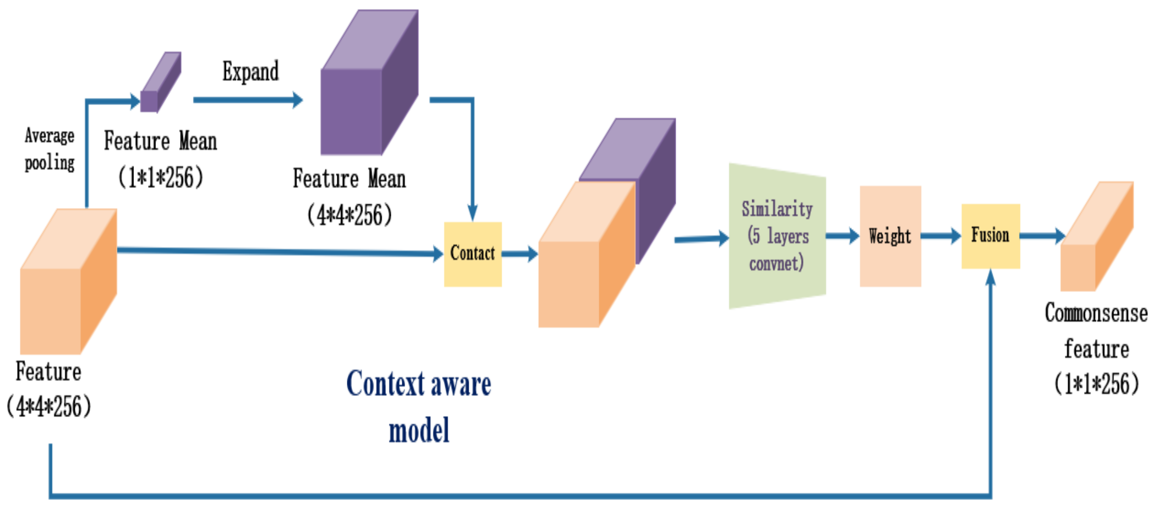
New heatmap

Original heatmap

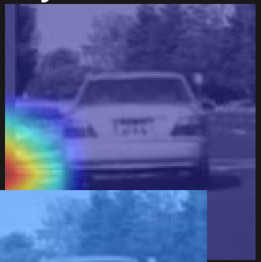
UISEE

You See Future

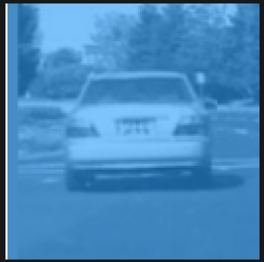
Commonsense feature



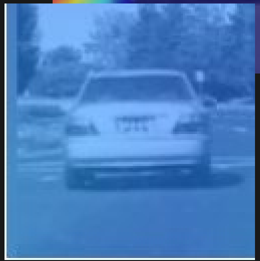
Template Similarity \weight



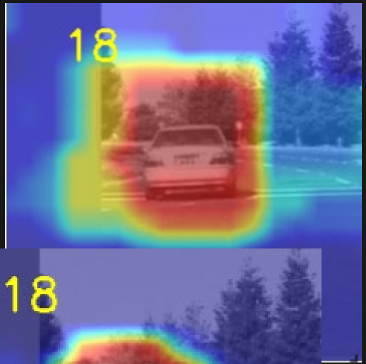
Template Commonsense feature



Search Original feature



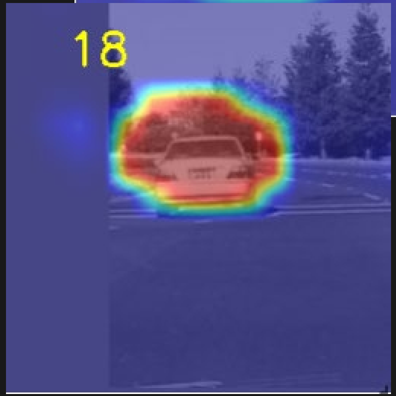
New heatmap



Search Similarity weight

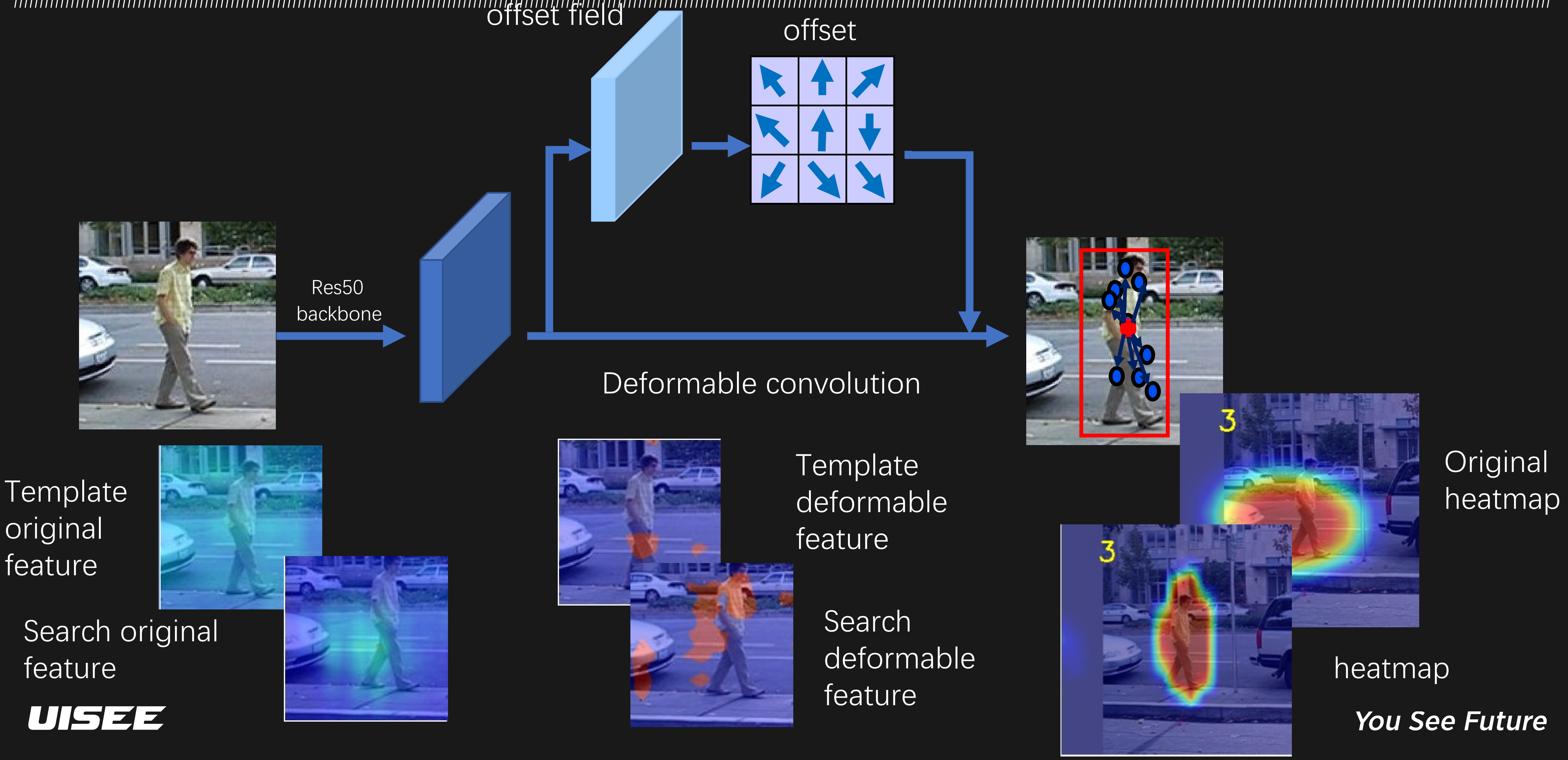
Search Commonsense feature

Original heatmap



Search Original feature

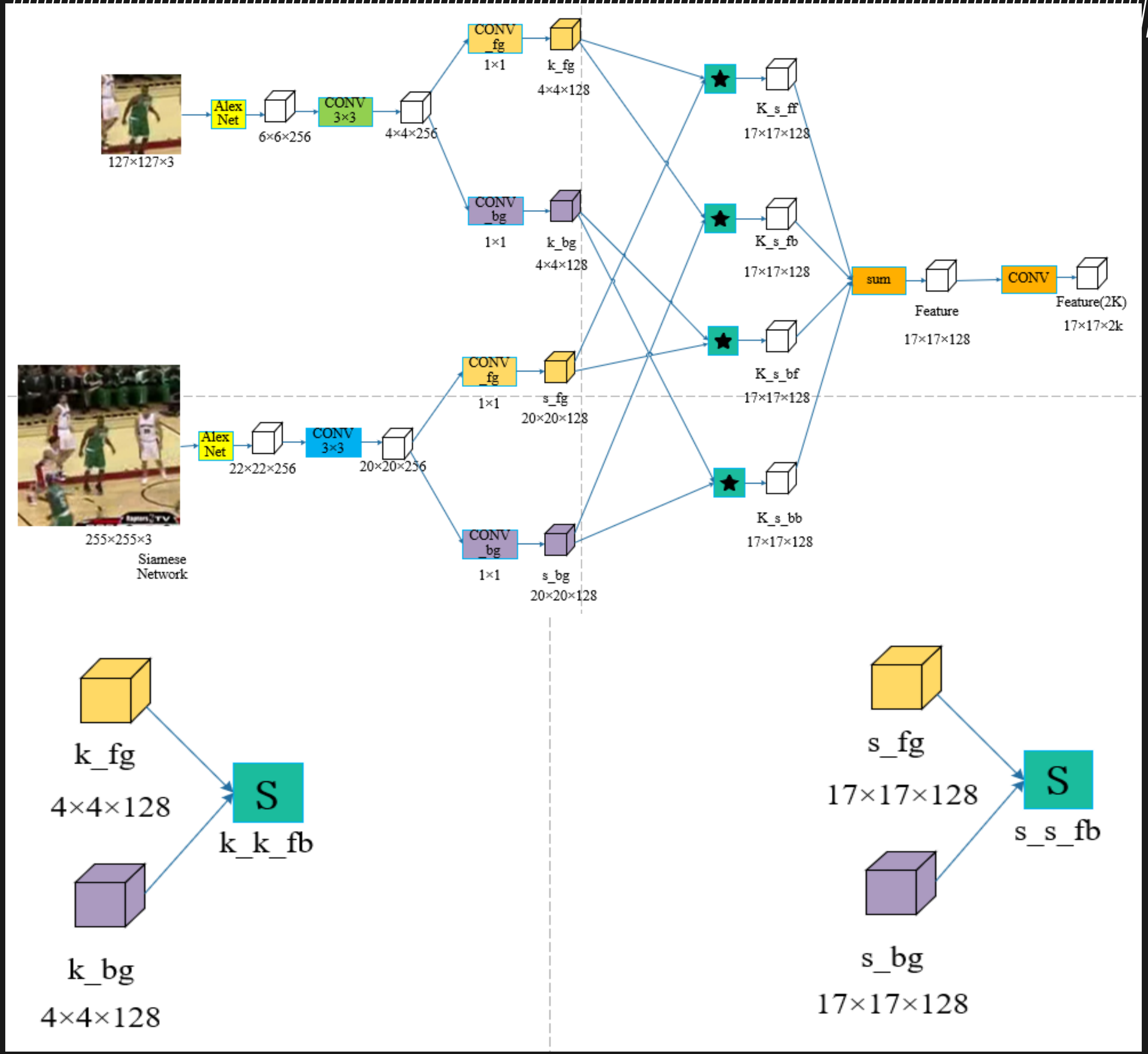
Deformable module to replace the common sense



UISEE

You See Future

Disentangle the feature into foreground and background



First , the extracted features are split into foreground features and background features and use inner product=0 to constraints the orthogonal disentangle.
Second, penalty the foreground and background feature by similarity loss.

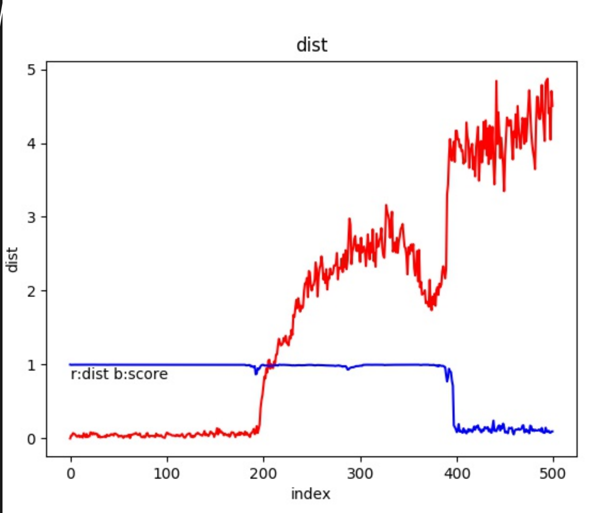
As the figure shows :

Bottom-left : background

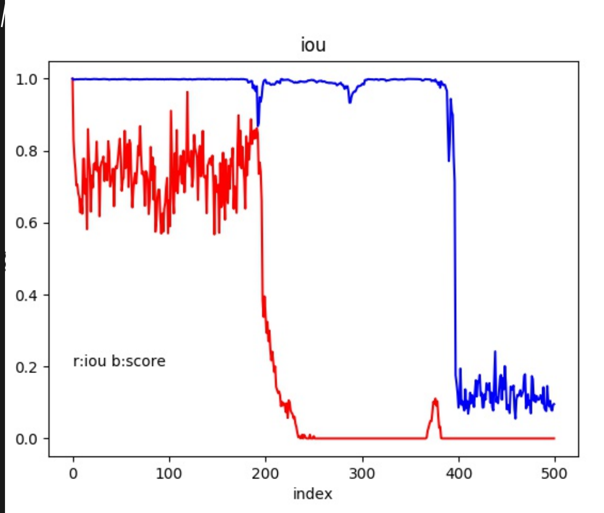
Bottom-right : foreground

The bg-branch and fg-branch learn different features

Disentangle model



R:dist B： score



R:iou B:score

As the figure shows :
Bottom-left : background
Bottom-right : foreground
The bg-branch and fg-branch learn different features

OTB 100

VOT

Deformable convolution

Tracker name	Success	Norm Precision	Precision
dcn	0.389	0.000	0.518

Tracker Name	Accuracy	Robustness	Lost Number	EAO
dcn	0.540	1.692	363.0	0.105

commonsense

Tracker name	Success	Norm Precision	Precision
commonsense_alex	0.399	0.000	0.536

Tracker Name	Accuracy	Robustness	Lost Number	EAO
commonsense_alex	0.549	1.659	356.0	0.106

Train rpn by pretrained model

Tracker name	Success	Norm Precision	Precision
train_rpn	0.377	0.000	0.500

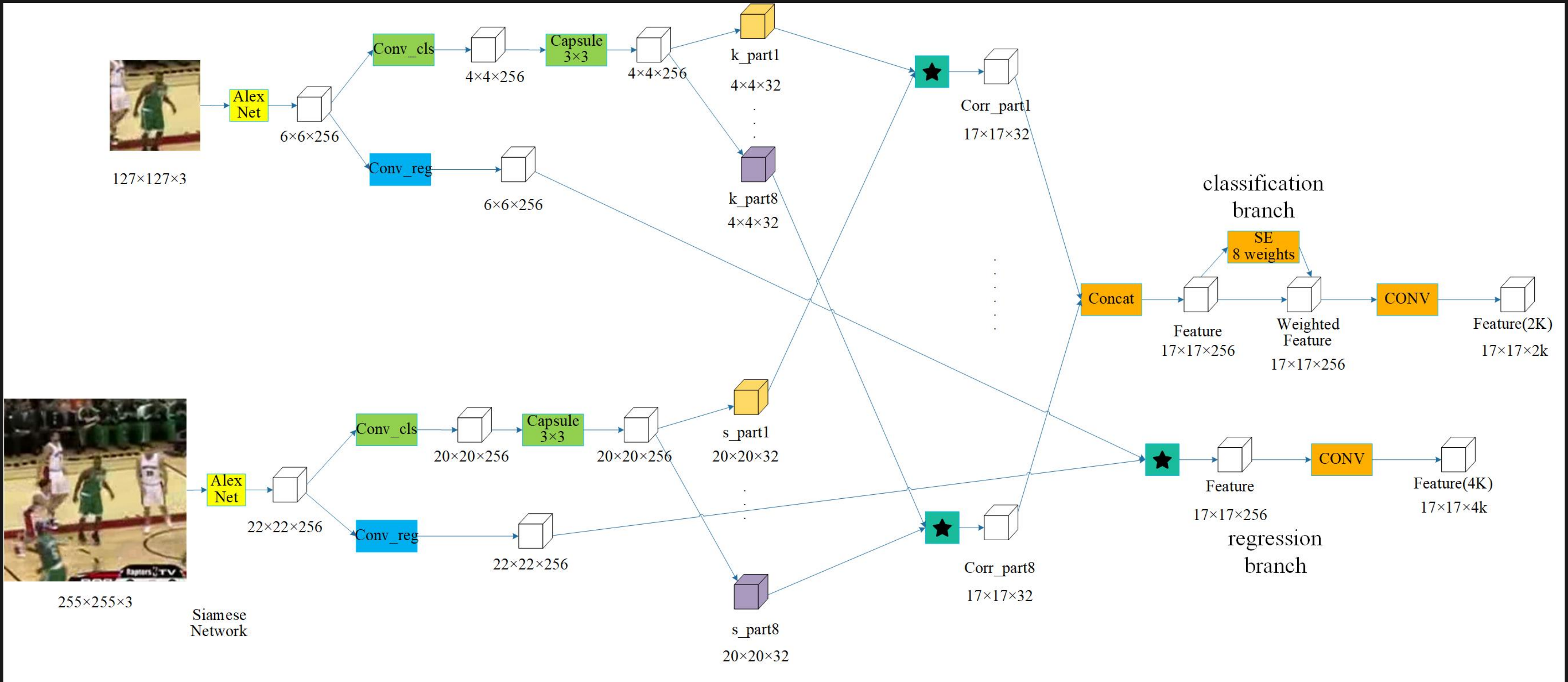
Tracker Name	Accuracy	Robustness	Lost Number	EAO
trainrpn	0.533	1.664	357.0	0.104

pretrained model

Tracker name	Success	Norm Precision	Precision
pretrain_rpn	0.416	0.000	0.555

Tracker Name	Accuracy	Robustness	Lost Number	EAO
pretrain_rpn	0.599	1.645	353.0	0.116

Disentanglement with Capsule



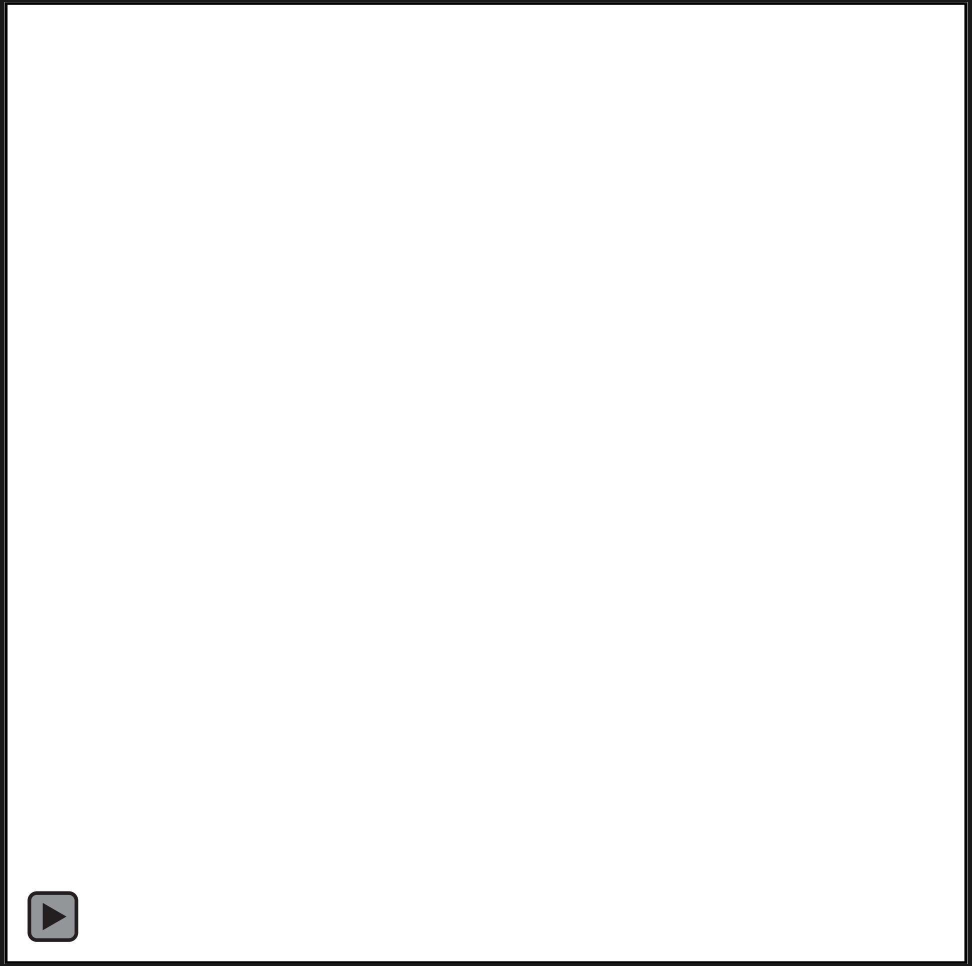
Multiple Part Disentangling

Move capsule out of the regression branch

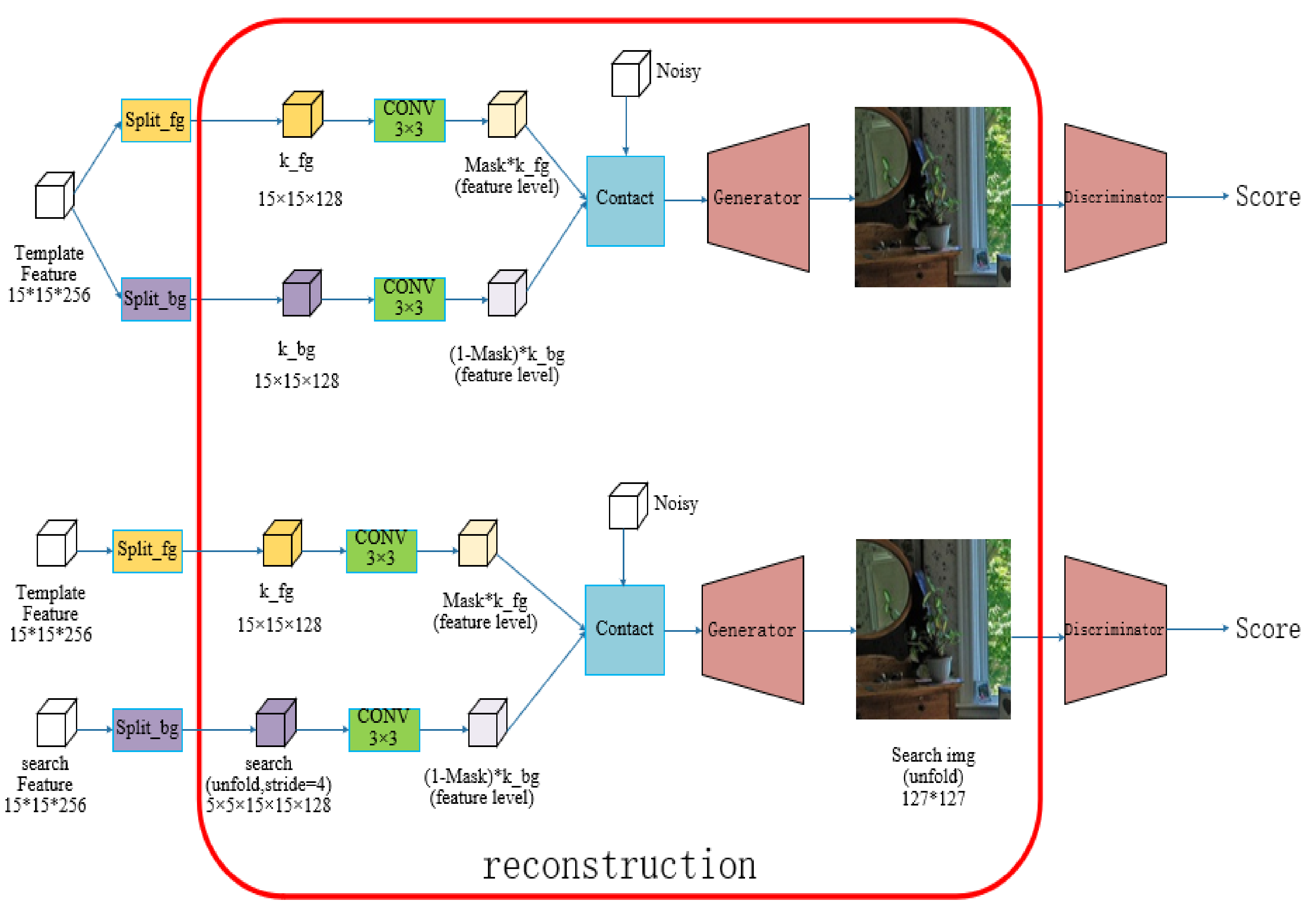
1, better overall performance on test dataset

2, classification: too large heatmap
regression: malfunctioning in test phase

3, Why perform well on validation set?
Maybe capsule needs more data to generalize to new dataset



Disentangle and Reconstruction model details

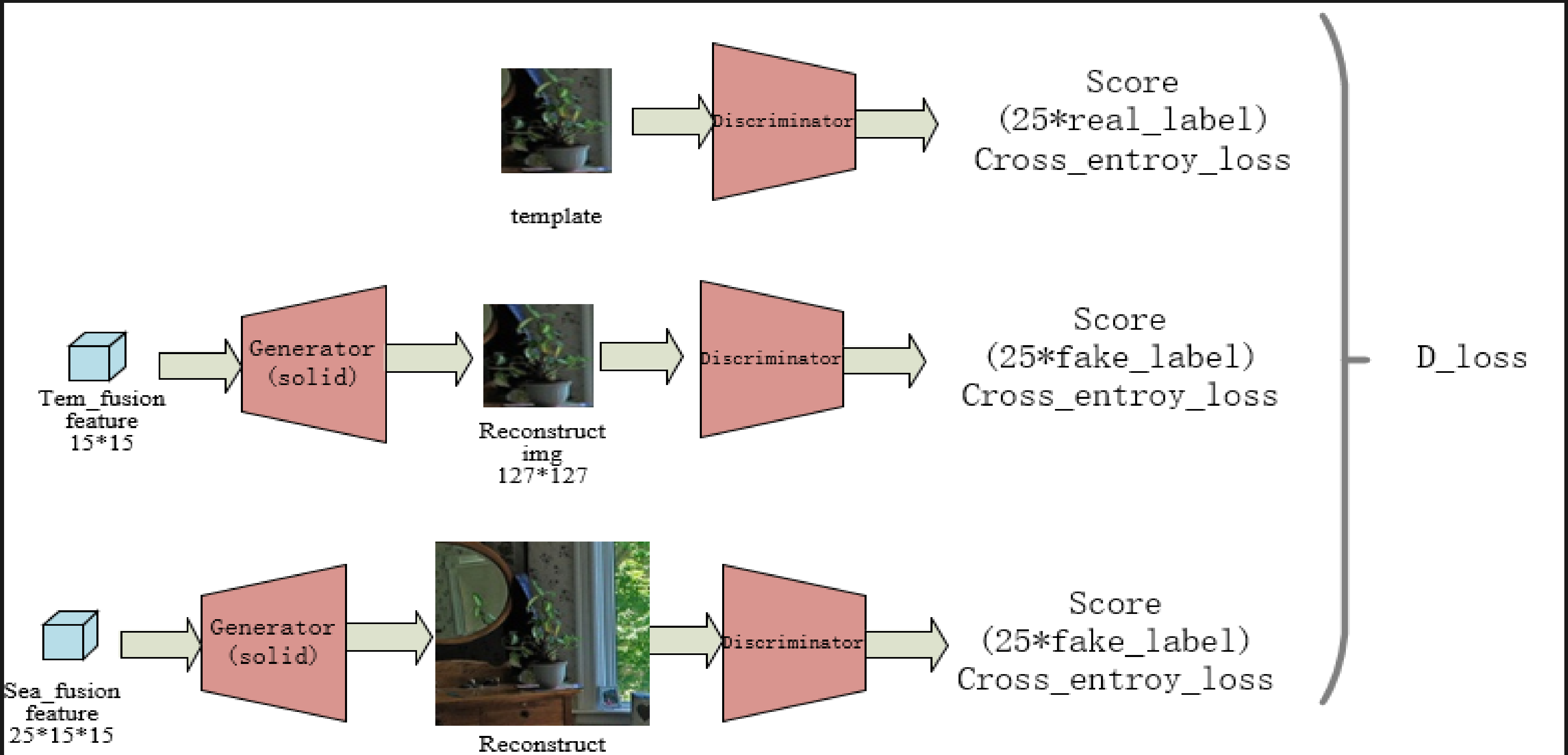


Train disentangle model and reconstruction mode and GAN:

top: after splitting template foreground and background, we contact them(with a learned mask) with noisy to reconstruct fake template image by the generator, then get a score by the discriminator

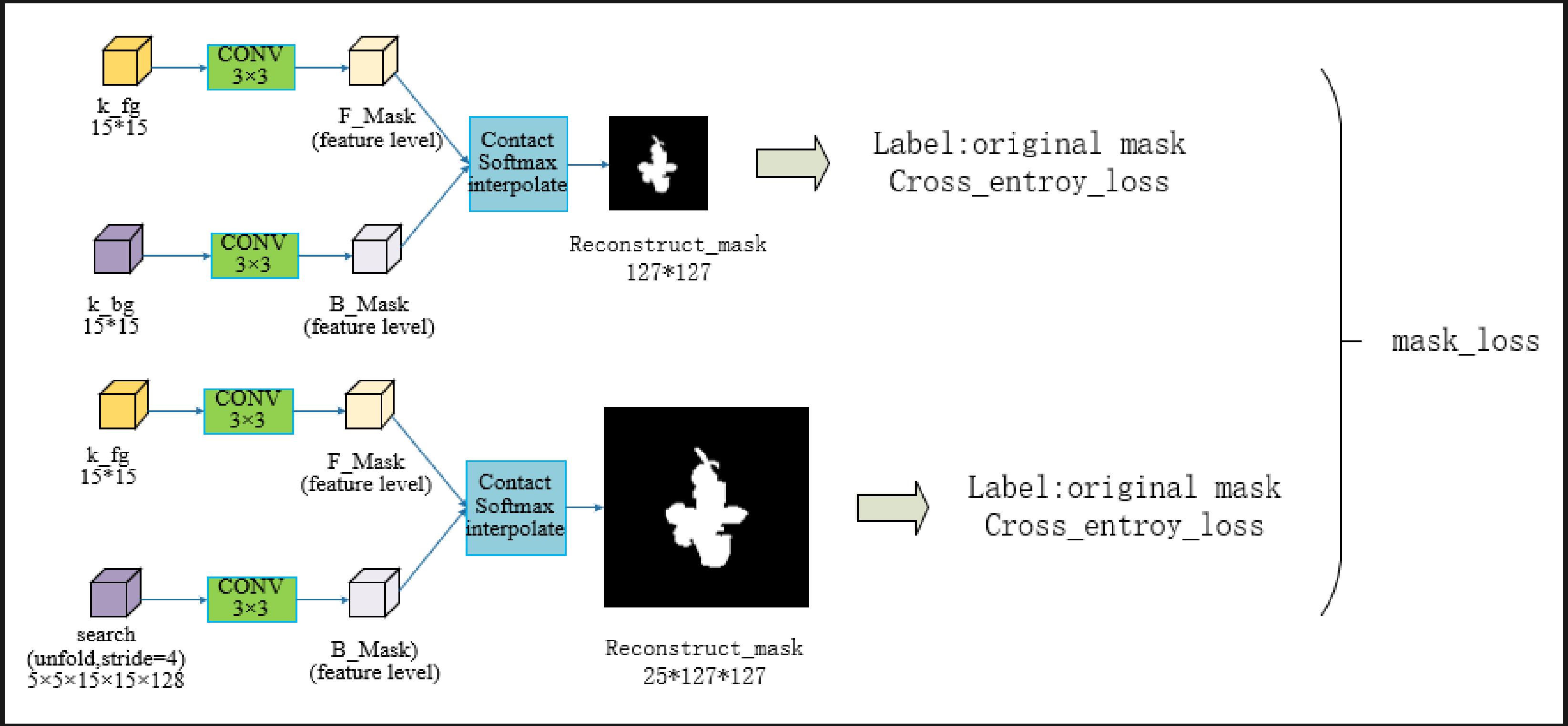
bottom: fusion template foreground feature and search background feature(with mask) to reconstruct fake search image, then get a score by discriminator

Loss details



After about 100 train images,D_loss has fallen below 0.1,discriminator has been invalid

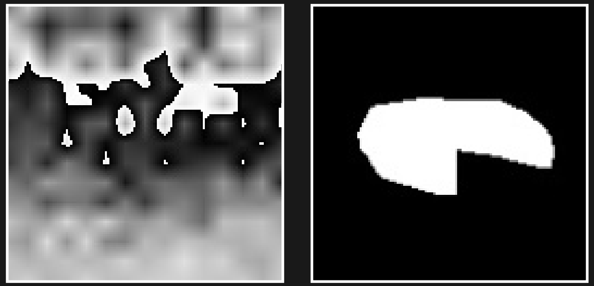
Loss details



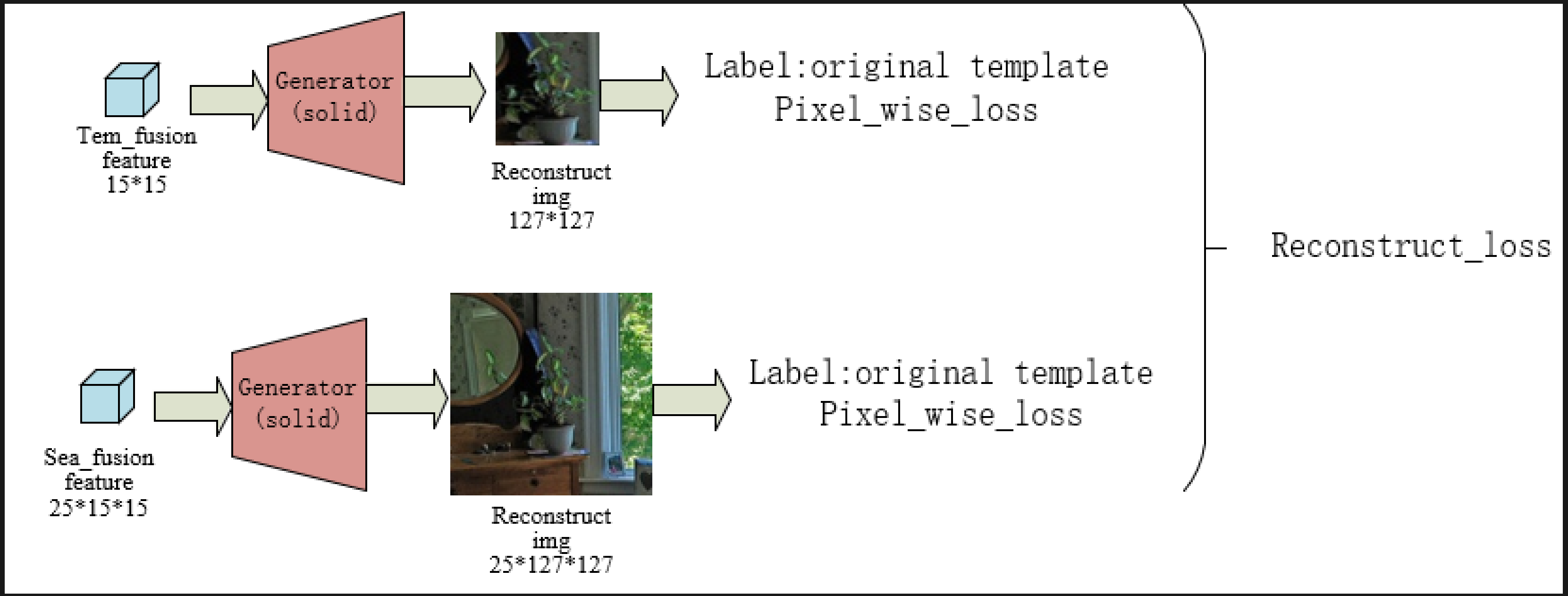
Mask_loss: we use classification loss to learn mask by view the original mask as label,but during the training process ,mask_loss is in a state of shock

Reason: too many softmax

1000 train images:



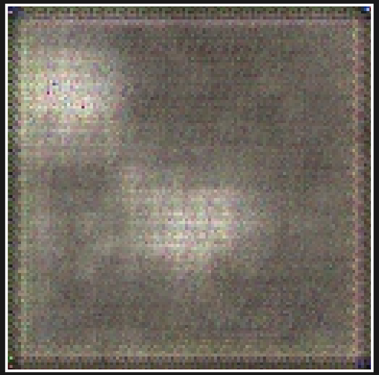
Loss details



reconstruct_loss: we use pixel_wise_loss to learn the reconstruction of image , and the reconstruct_loss is in a falling state

2000 train images:

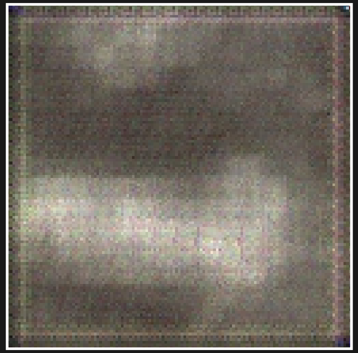
Reconstruct_img



Original_img

3000 train images:

Reconstruct_img



Original_img