Study of filtering the negative effects of weather on the detection of objects

Stadnik A.V.

ОЦРВ

Problem formulation

Task of working-time standarts

- The business process of developing time standards
- Manual processing
- Need to automate process

Basic models and algorithms

Detection task

- People and tools
- Keypoints
- Human-object Relations (HOI)

System architecture

Storage service

Pre-processing | Human detection | Tool detection | HOI

Tasks

Pre-processing	Human detection		Tool detection	HOI	
Reduction to the processing size and splitting into frames	Yolov5 human detection	Keypoints detection by mediapipe or Yolov7-pose	Determination by a neural network model of the type, position and size of a set of 36 tools	Determining whether the tool is in the hand. for all people, every hand, all tools	Time tracking

36 tools

Task-control service

analysis of working hours divided into operations performed

Research directions

Development of the model architecture

- Improved detection metrics
- Increasing the robustness of the model
- Input data filtering methods

Development of the methodology for determining the technological operation

- Objects of influence
- The nature of movements - recognition of actions

Tracking employees on video

- Identification based on embeddings
- Identification based on key points
- MOT - tracking

Increasing the robustness of the model - stylization

Stylization changes the visual perception of the image

The color content and textures change, the content remains

The generalizing ability of the model is improved

Accuracy on real data is increasing

+

content

Increasing the robustness of the model - weather

Weather phenomena greatly distort the scene

Weather augmentation in the training of detection models

The generalizing ability of the model is improved

Accuracy on real data is increasing

+

accuracy

Input data filtering methods

Unfavorable shooting conditions (snow, rain)

Filter the image - reduce the negative effect

Neural network models of rain and snow filtration

Improving the quality of detection of small instruments

Filtering

#

patterns

Input data filtering methods - models

Different types of neural network models

Different preprocessing results

The recurrent model filters heavy rain

Encoder-decoder filters light rain

1!=2

different models gives different result

Weather conditions - how to deal with?

Weather detection

Classic CV

Rain, Snow

Fog

/Dl/

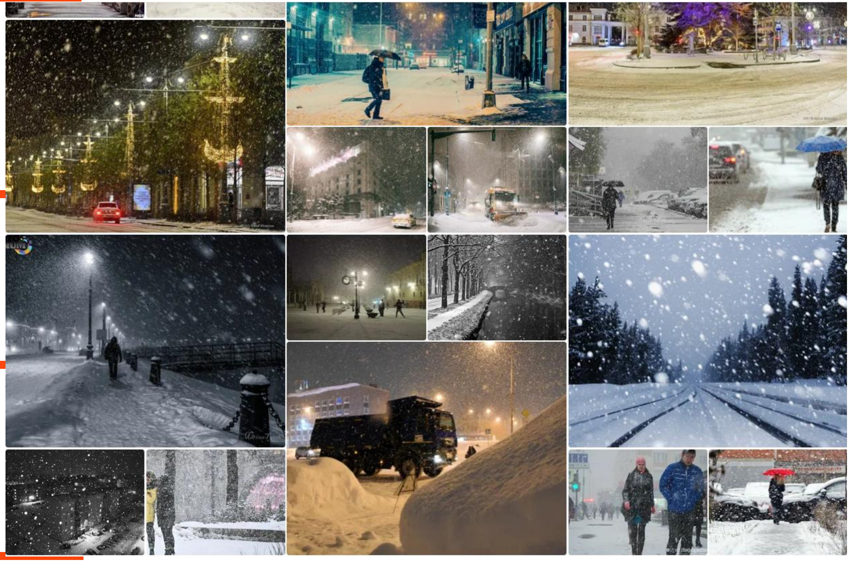
Deep model classifier (Resnet-like)

Rain, Snow, Fog

Weather detection: /DL/approach

Datasets:

- DAWN Dataset
- Weather Image Recognition @Kaggle
- Custom images



Weather detection: /DL/approach

Resulting Dataset

Combined & Filtered

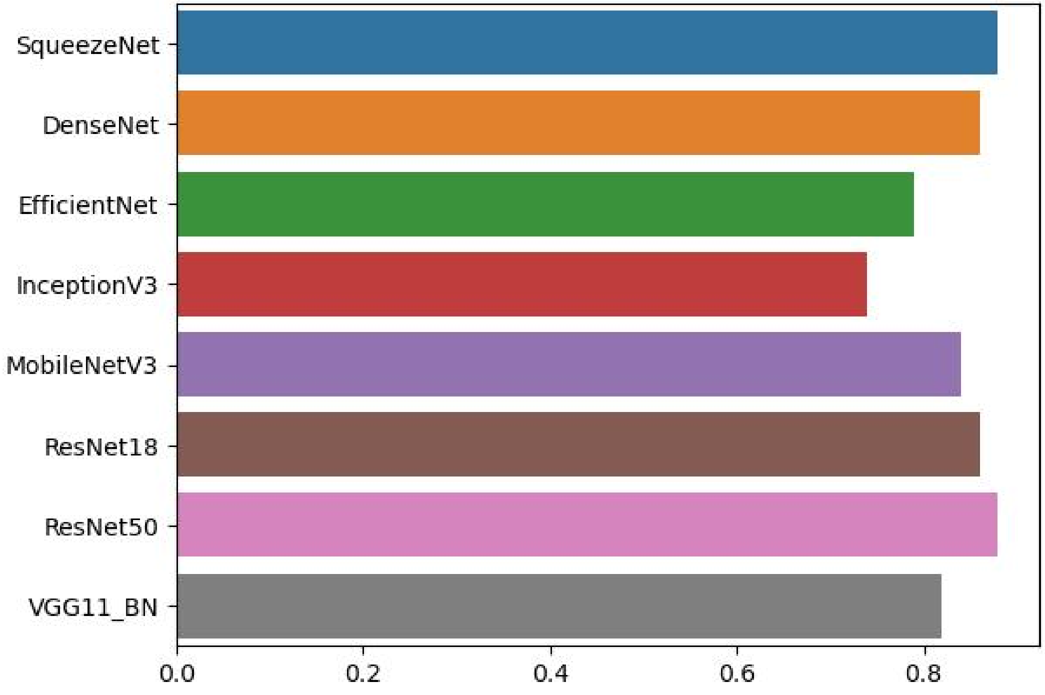
Fog (1407), Norm (1605), Rain (1124), Snow (543)

Snow = Snowfall
only dynamic here

DL-classifiers comparison

- SqueezeNet: 0.88
- DenseNet: 0.86
- EfficientNet: 0.79
- InceptionV3: 0.74
- MobileNetV3: 0.84
- ResNet18: 0.86
- ResNet50: 0.88
- VGG11_BN: 0.82

Model comparison



DL-classifier

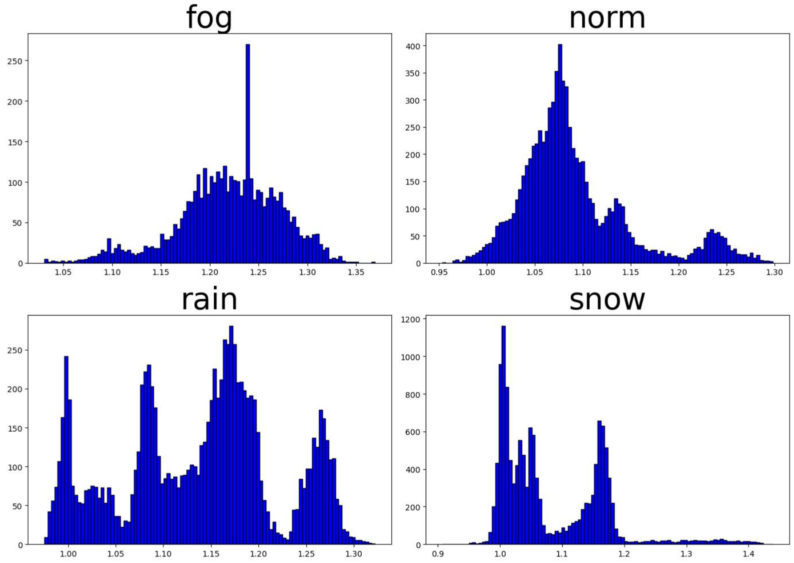
Choice for inference: MobileNetv3
Val & Test metrics

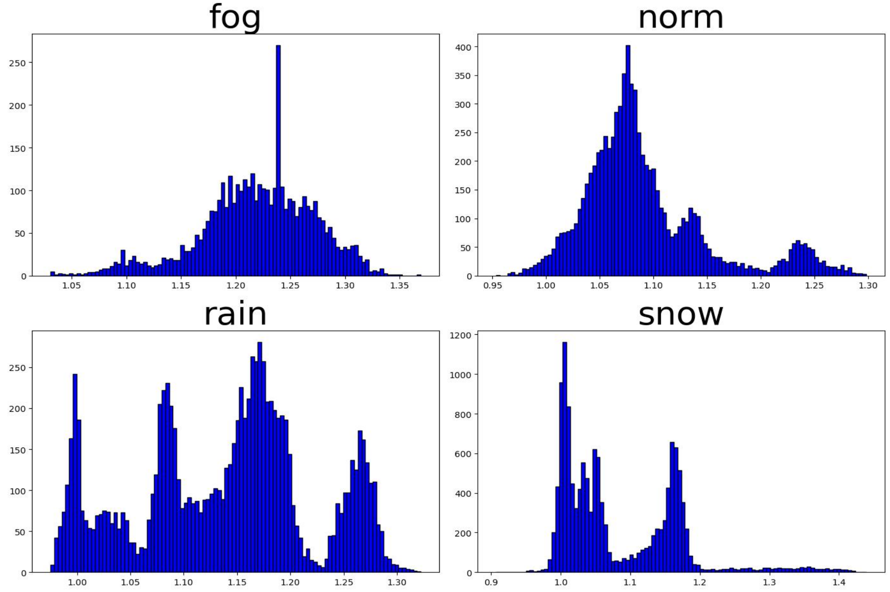
	Precision	Recall	F1-Score	Support
Fog	0.922360	0.947368	0.934697	627.000000
Norm	0.802632	0.938462	0.865248	650.000000
Rain	0.851030	0.859200	0.855096	625.000000
Snow	0.969163	0.810811	0.882943	814.000000
Accuracy	0.884021	0.884021	0.884021	0.884021
Macro Avg	0.886296	0.888960	0.884496	2716.000000
Weighted Avg	0.891319	0.884021	0.884248	2716.000000

	Precision	Recall	F1-Score	Support
Fog	0.956113	0.947205	0.951638	644.000000
Norm	0.831117	0.941265	0.882768	664.000000
Rain	0.858506	0.901503	0.879479	599.000000
Snow	0.955524	0.823239	0.884462	809.000000
Accuracy	0.898748	0.898748	0.898748	0.898748
Macro Avg	0.900315	0.903303	0.899587	2716.000000
Weighted Avg	0.903852	0.898748	0.898877	2716.000000

Classic CV

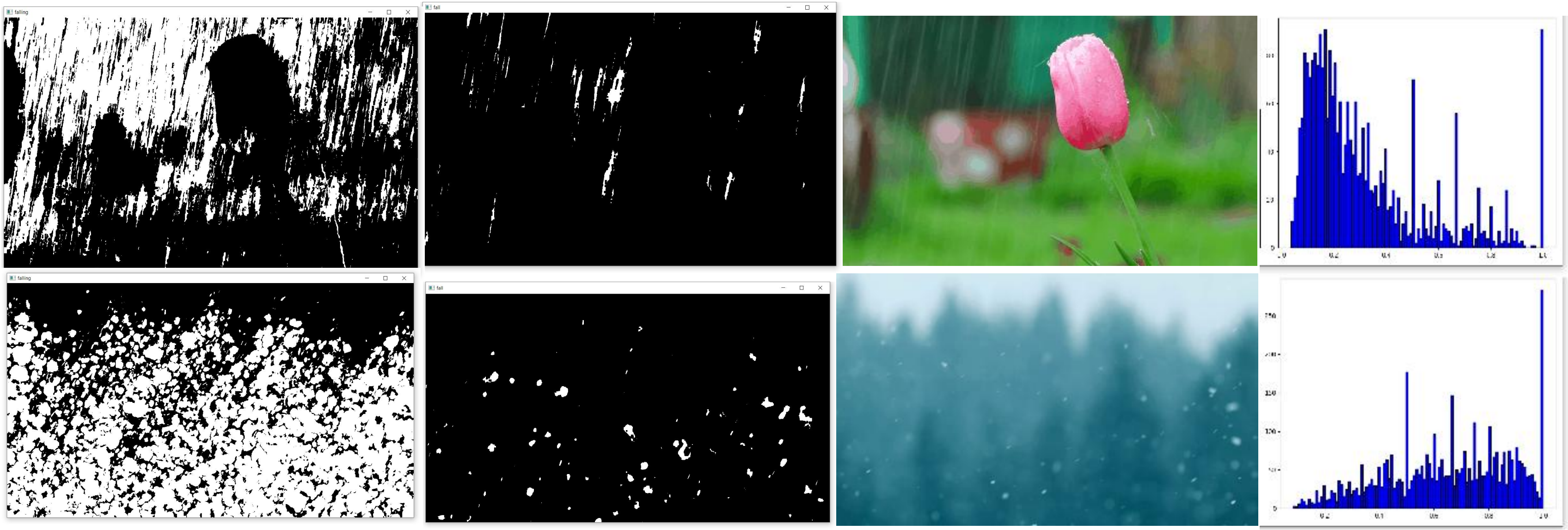
Search criteria - sum of wavelet details + PSNR(signal/noise)
is_psnr = psnr > 1.15 && is_wdd = wave_dd < 1.0
&& is_entropy = entropy > 5.5 and entropy < 7.7





Rain & Snow

Masks of difference frames and statistics for 50 frames.



Rain & Snow

Problem: non-stationary background.
Solution: optical flow mask
(pi/8~20 degree angle)

Image cleaning (derain)

Rain model: $y = x + r$: noisy element - rain(snow, fog)
The “encoder-decoder” architecture (U-Net) is applicable in most recovery tasks (low-level + high-level features).
In the task (rain) - small details => the perception area of the filter is small.

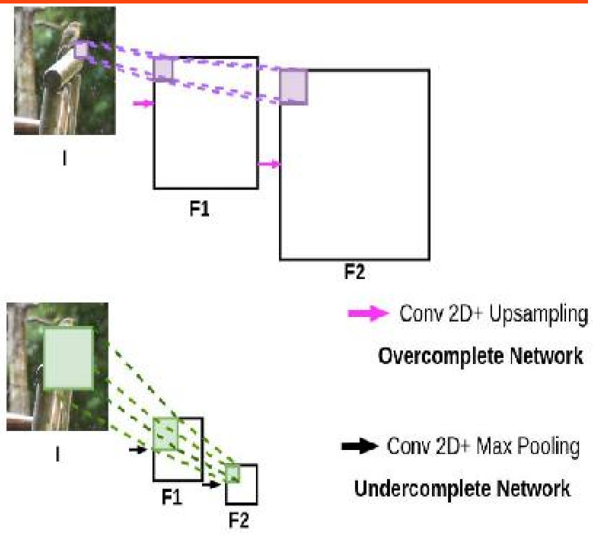
Image cleaning (derain)

OUC-D: two branches. Idea: local and global signs at the same time.

overcomplete-autoencoder
undercomplete-autoencoder

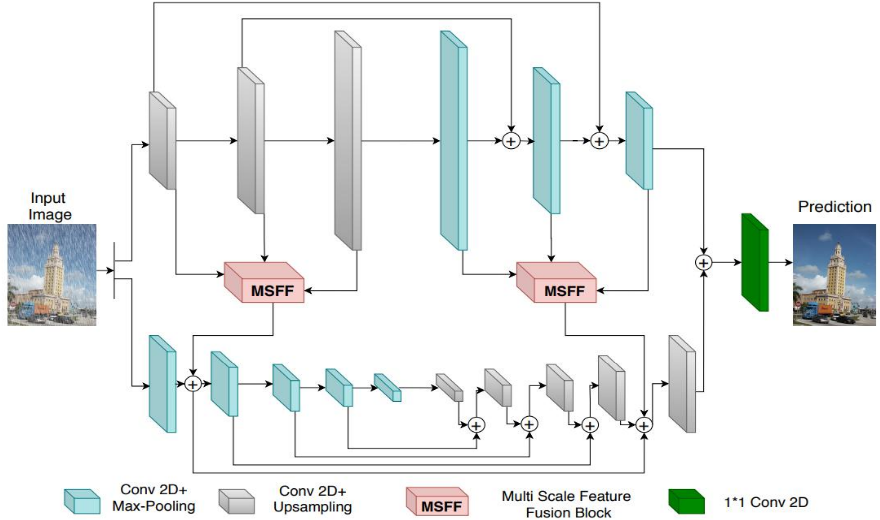
The task of maximum informativeness in local features

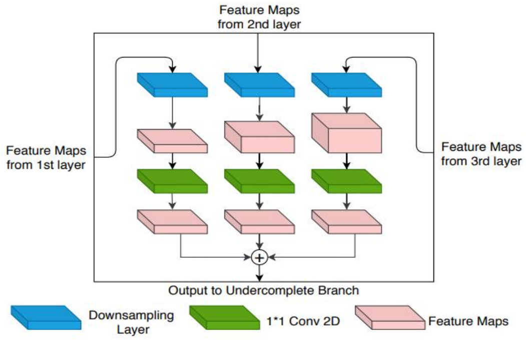
We do not exclude global signs - they contain significant information for recovery



Архитектура сети:

Перед добавлением карты объектов из ветки overcomplete - Multi-Scale-Feature-Fusion (MSFF) Block



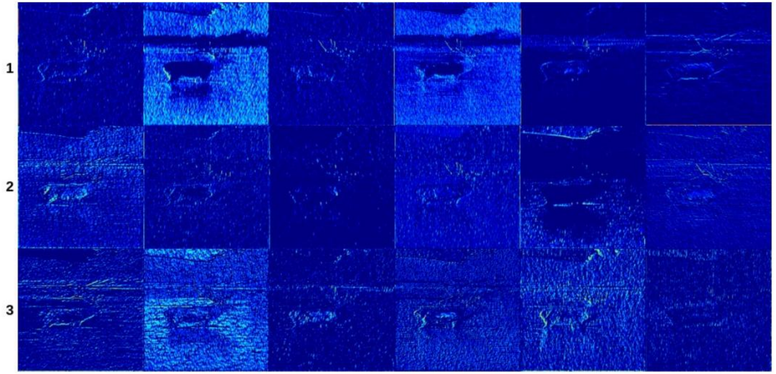


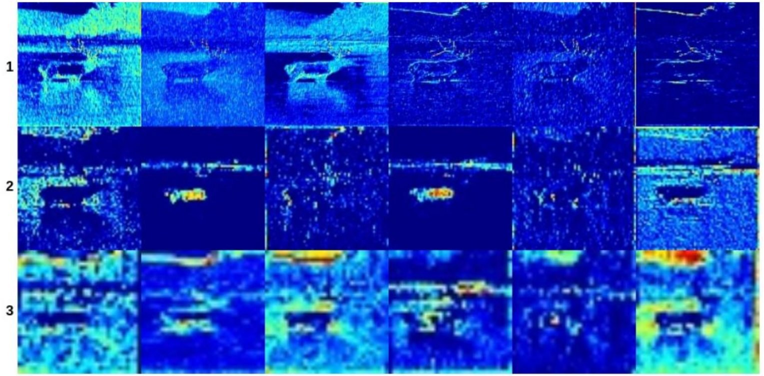
$$\mathcal{L} = \mathcal{L}_{mse} + \lambda \mathcal{L}_p, \qquad \mathcal{L}_{mse} = \|\hat{x} - x\|_2^2.$$

$$\mathcal{L}_p = \frac{1}{NHW} \sum_i \sum_j \sum_k \|F(\hat{x})^{i,j,k} - F(x)^{i,j,k}\|_2^2,$$

Perceptual loss (*J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” Springer, 2016)

Image cleaning (derain)





Features in the overcomplete architecture. The network captures local information - small details (rain bands and drops)

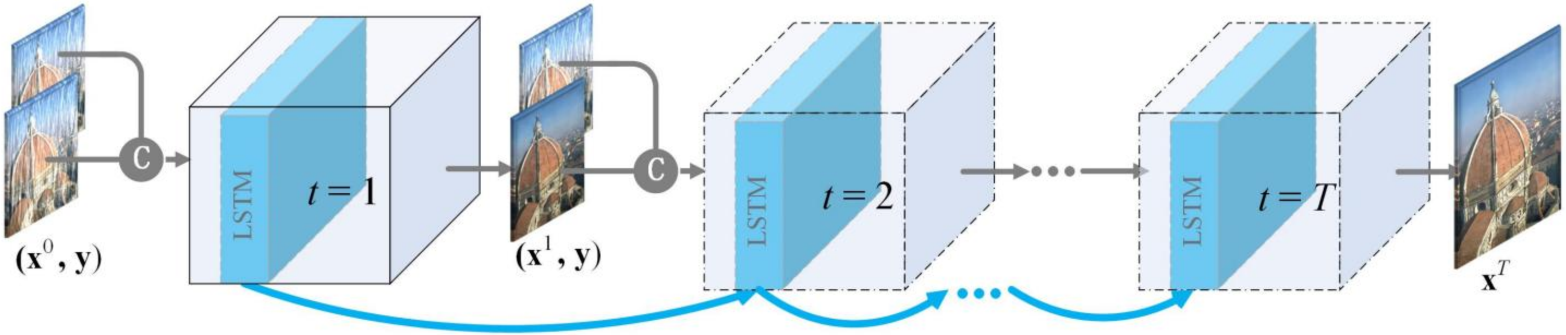
Featuremaps in undercomplete architecture. The network captures high-level information (animal and background)

OUCD-network: some results

*Exploring Overcomplete Representations for Single Image Deraining using CNNs https://arxiv.org/pdf/2010.10661.pdf

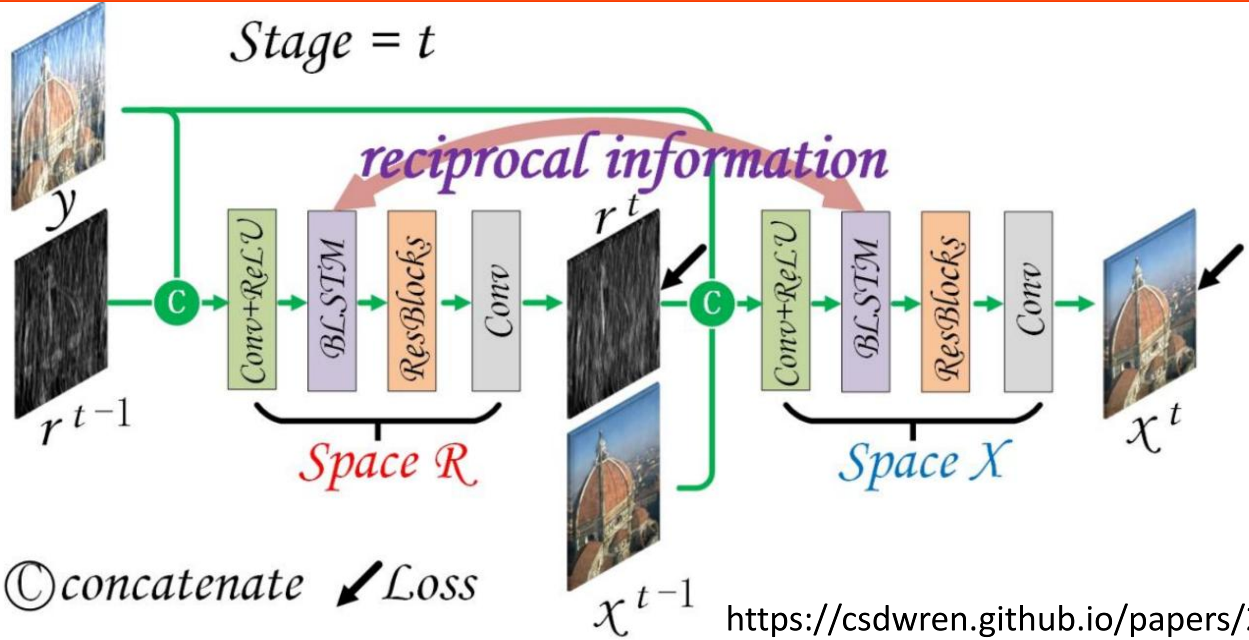
SRN-network

SRN is built on the basis of ResNet => recursively T times.
The complexity of training is reduced by splitting into several stages.
The ability to make F - computationally easier.
SRN network inference: X(t) = F(Y, X(t-1))
parameters F are reused



BRN-network

BRN-network deal with rainy image and rain mask



https://csdwren.github.io/papers/2020_tip_BRN.pdf
https://sigport.org/sites/default/files/docs/BRN_slides.pdf

BRN Image Filtering:Test Images

The proposed solution is
BRN-network:
corrects strong
distortions
frame-by-frame
operation
speed of operation
(in comparison with
simple filters)

BRN Image Filtering: unknown network augmentation

Similar to image “normalization”
Does not completely remove noise if patterns are unknown

BRN Image Filtering: Synthesized images

It can be concluded that the “concept” of rain in BRN and generative network coincide

Neural network filtering approach:

DerainZoo: a set of real and synthetic datasets (https://github.com/nnUyi/DerainZoo/blob/master/DerainDatasets.md)

OUCD – works on light rain

BRN – excellent in heavy rain

Input image with noise (rain, snow) - the output of the model is a clean image
The difference (PSNR, SSIM) between the input and output will allow you to assess the presence of rain (snow, etc.)

Filtration effect on detection

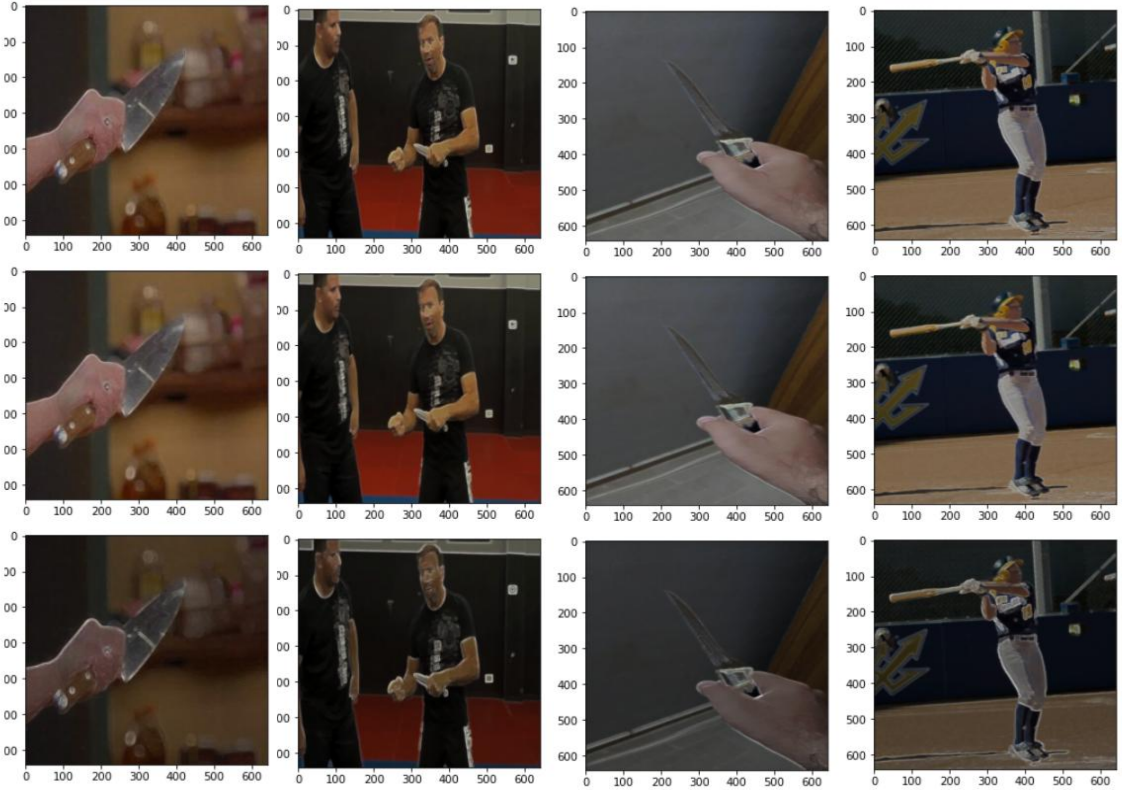
Motivation: improving the accuracy of object detection (small tools)
Snow + rain - high-frequency details: the filter will remove the high-frequency component of the signal.
evaluate how much detection changes during filtering;
add information about the “edges" and also evaluate the impact.
Network: Yolov5

Фильтрация

Dataset to check: 3 classes (bat, pistol, knife) 3000 images (https://universe.roboflow.com/ntut-zy5y0/weapon-detection-yfvuq)

Dataset - changes

From top to bottom:
original image
after median filter
filter + edges (sobel)



Results: preliminary

- ###original
- Class P R mAP50 mAP50-95
- all 0.63 0.546 0.584 0.24
- baseball-bat 0.553 0.492 0.503 0.205
- knife 0.663 0.725 0.737 0.326
- pistol 0.676 0.422 0.512 0.189
- ### median filtered
- Class P R mAP50 mAP50-95
- all 0.632 0.489 0.552 0.235
- baseball-bat 0.546 0.439 0.484 0.198
- knife 0.793 0.701 0.755 0.346
- pistol 0.557 0.328 0.417 0.16
- ### median filtered + Sobel edges
- Class P R mAP50 mAP50-95
- all 0.631 0.538 0.573 0.237
- baseball-bat 0.574 0.483 0.488 0.182
- knife 0.735 0.722 0.769 0.354
- pistol 0.585 0.409 0.464 0.176

The effect of BRN-filtering on object detection

The task is to detect tools
Problem: weather events reduce the quality of detection, especially of small instruments

Solution: frame filtering
The dataset for verification contains large, medium and small objects

Filtering + object detection - evaluation methodology

Let’s indicate model currently used as “standard” - std, Filtering as F, augmentation of weather as aug

Validation set val, and with augmentation: val+ = aug(val)

Filtering can be different:
F = [med, sob, med+sob, emboss, BRN]

(std+F)(x) = (std(F(train)))(x)

Some questions & notations

What is actually changing? std(val) <-> std(F(val))

And if it's “rain"? std(val) <-> std(val+)

If we remove the “rain"? std(val+) <-> std(F(val+))

Do we need filtering? (std+out)(val), (std+out)(val+)

Does F help? (std+aug)(F(val)), (std+aug)(F(val+))

”Glasses" from bad weather (std+F)(F(val)), (std+F)(F(val+))

Filtering effect to object detection task

***notation: (P, R, [mAP50, mAP50-95])**
**** for small obj (knife)**

What is actually changing? std(val) <-> std(F(val))
(0.66,0.72,[0.73, 0.32]) → (0.68,0.69,[0.74 ,0.31])

And if it's “rain"? std(val) <-> std(val+)
(0.66,0.72,(0.73, 0.32)) → (0.43,0.20,[0.20,0.07])

If we remove the “rain"? std(val+) <-> std(F(val+))
(0.43,0.20,(0.20,0.07)) → (0.67,0.62,[0.65,0.25])

Filtering effect to object detection task

***notation: (P, R, [mAP50, mAP50-95])**
**** for small obj (knife)**

Do we need filtering? (std+out)(val), (std+out)(val+)
(0.69,0.68,[0.69,0.29]), (0.69,0.68,[0.69,0.29])

Does F help? (std+aug)(F(val)), (std+aug)(F(val+))
(0.66,0.57,[0.63,0.28]),(0.81,0.69,[0.75,0.32])

Filtering effect to object detection task

***notation: (P, R, [mAP50, mAP50-95])**
**** for small obj (knife)**

”Glasses" from bad weather (std+F)(F(val)), (std+F)(F(val+))
(0.76,0.69,(0.78,0.35)), (0.80,0.70,(0.73,0.30))

This option gives the best metrics:

(std+F)(F(x)) > (std+aug)(x)
(std+F)(F(x+)) ~ (std+aug)(F(x+))

Filtering effect to object detection @ ASUTR

F = BRN (rain + snow)-> improvements in small instrument detection metrics.

'+chisel'
'+hammer'
'-pliers'
'-wrench'
'+scraper'
'--brush'
'+paw_for_gasket'
'=handaxe'

class	precision	recall	map05	map05-95
chisel	-0,058	-0,008	-0,021	-0,053
hammer	-0,015	0,001	-0,008	-0,009
scraper	-0,063	0,020	-0,018	-0,020
tie_tong	-0,036	-0,033	-0,034	-0,023
brush	0,039	0,021	0,032	0,026

ОЦРВ

Thanks for your attention!

Corrections! ?
Suggestions?
Remarks(?
Questions ??