Blind Preprocessing

To Default viola jones

Pre-Processing

DataSets : Desired Proportion

1. Yale
2. BioID
3. MIT CBCL
4. Caltech
5. Sof
6. Non face

detectMultiScale : internals

- To get the results of cascade classifier with Level weights and Confidence
- 3 methods:
  - detectMultiScale
  - detectMultiScale2
  - detectMultiScale3
- faceCascade.detectMultiScale3 : 25 stage Cascade Classifier
- Returns:
  - Number of faces detected
  - Bounding box of each face detected
  - Confidence of each detection (sort of decision score at last Level Classifier)
  - RejectLevels (in earlier versions used to return the level at which Classifier rejected as non-face , but recent updates in Opencv made it return just the last level number if image pass all levels of Cascade classifier)
- Call signature:

```
faces, reject_levels, level_weights = classifier.detectMultiScale3(gray_img,
scaleFactor=1.0485258,minNeighbors=min_neighbors, outputRejectLevels=True)
```

minNeighbors: 3 or 5

- Both

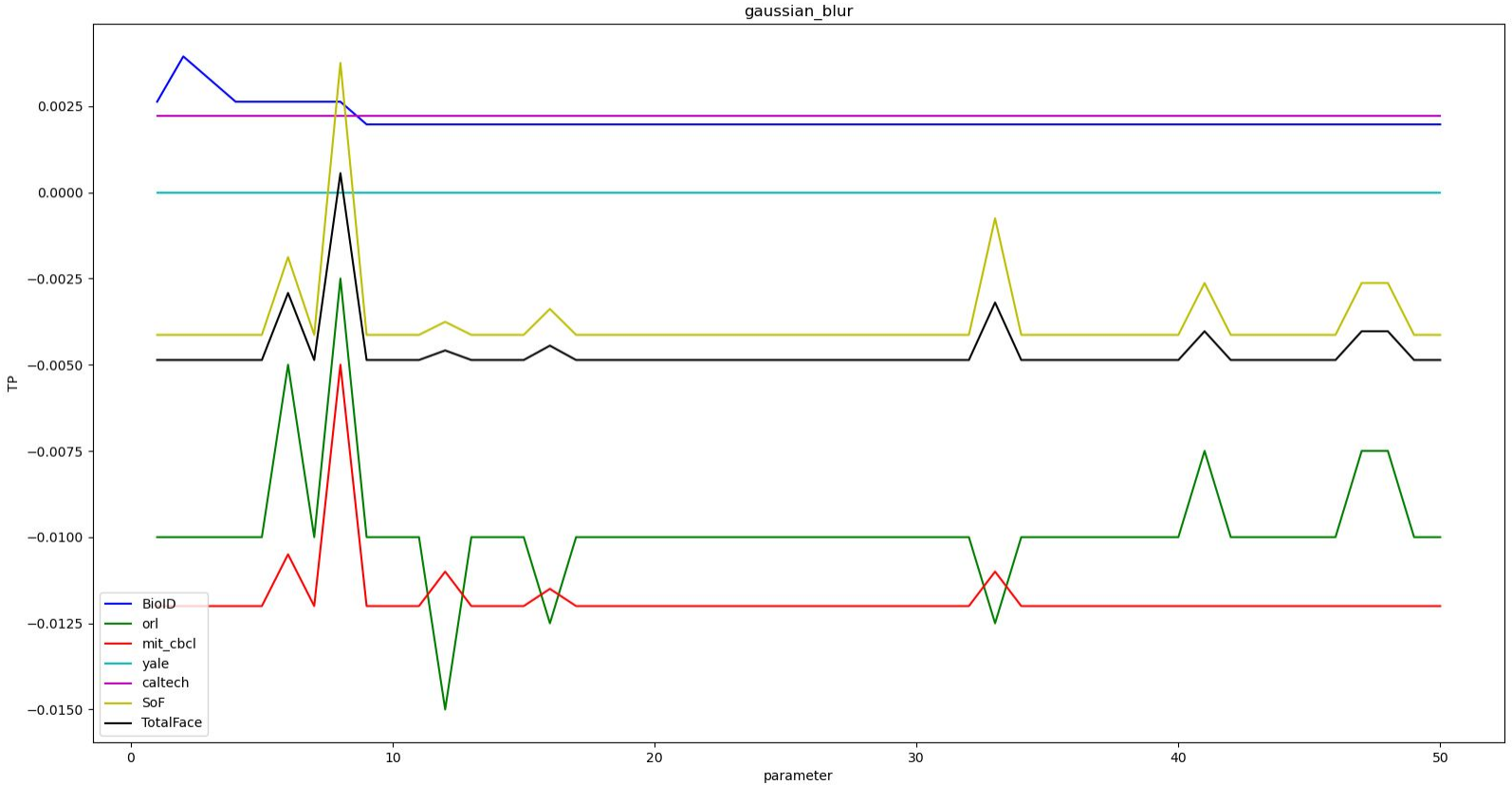
Plotting results for 7 Datasets (only face datasets)

- Difficulty levels:
  - Yale : Easy
  - SoF : Difficult
- 2 other datasets:
  - TotalFace: Result of 6 Face Datasets combined
  - NonFace: Result of 8 Non-Face Datasets Combined

1. Gaussian Blur

True Positive graph

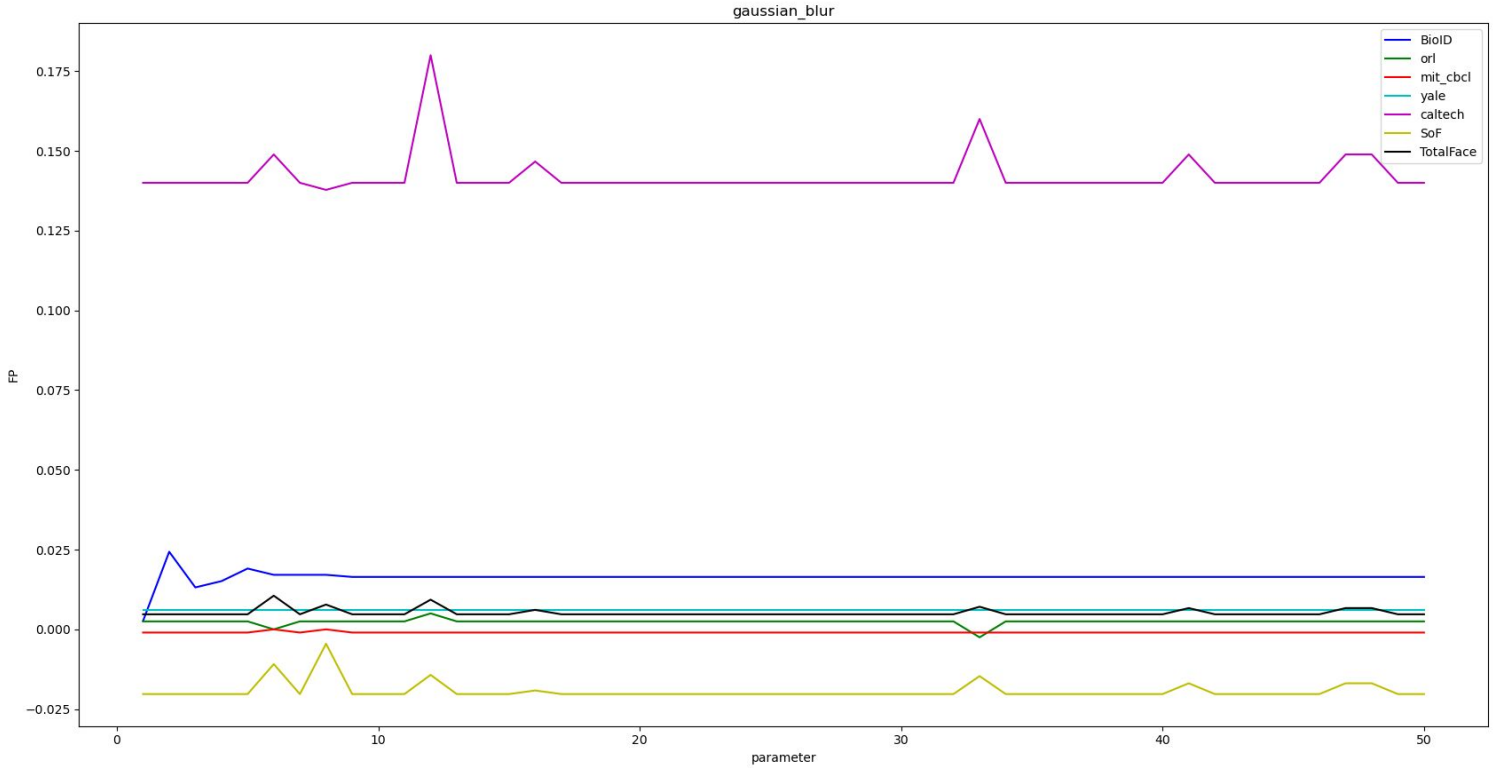
Kernel size = 5 (fixed) Sigma = 1 to 50



Sigma = 8 (seems good)

FP graph

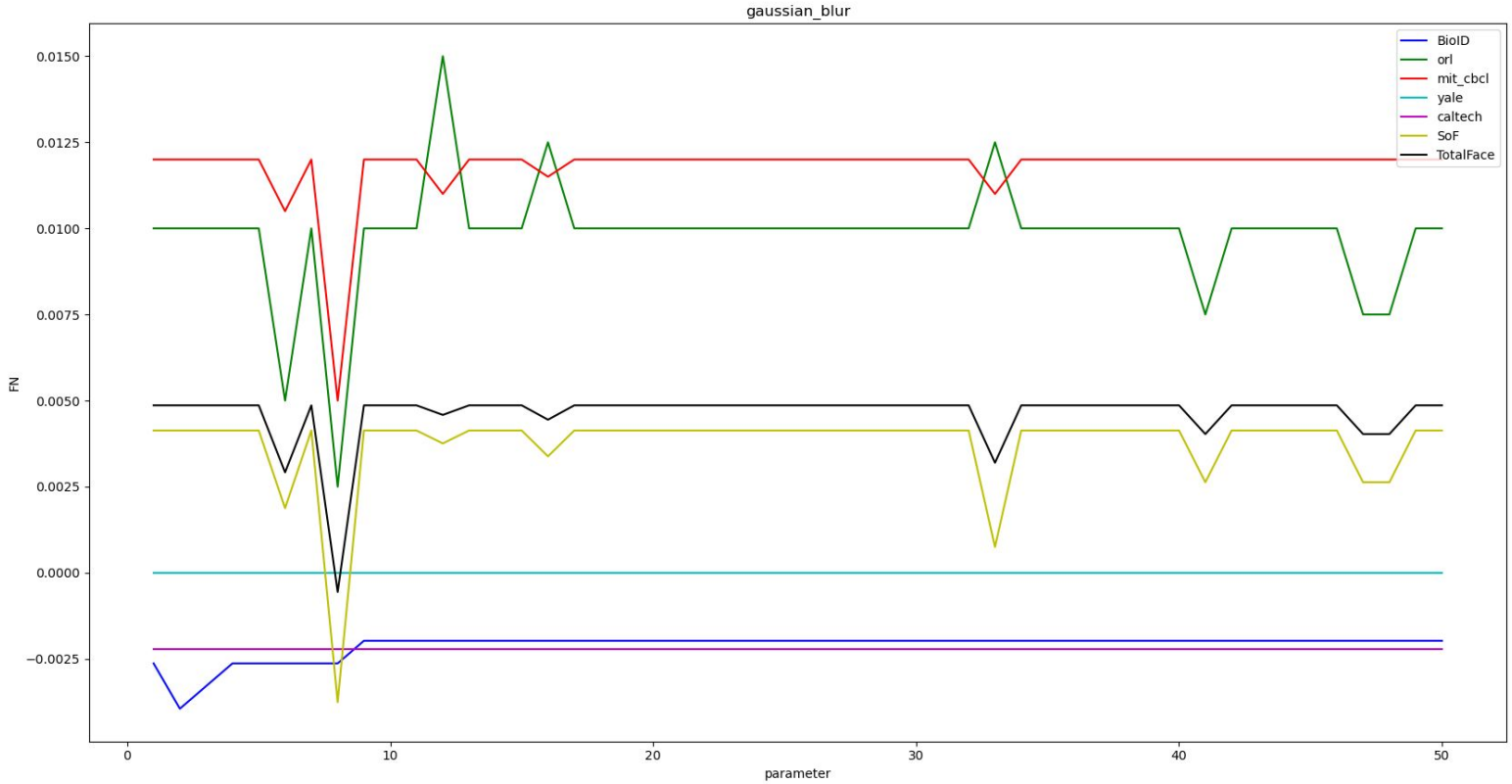
Kernel size = 5 (fixed) Sigma = 1 to 50



Sigma = 8 (seems good)

FN graph

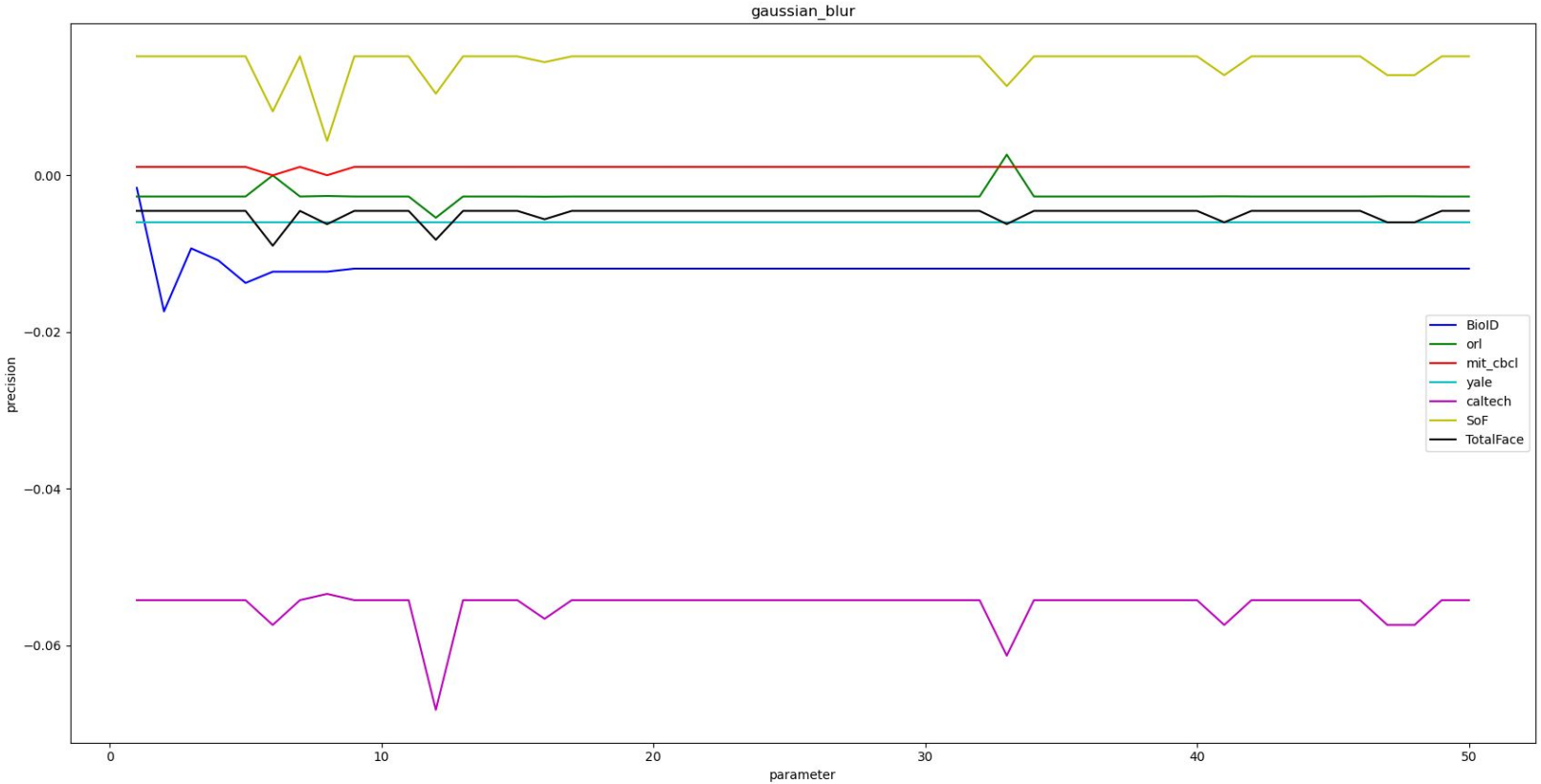
Kernel size = 5 (fixed) Sigma = 1 to 50



Sigma = 8 (seems good)

Precision graph

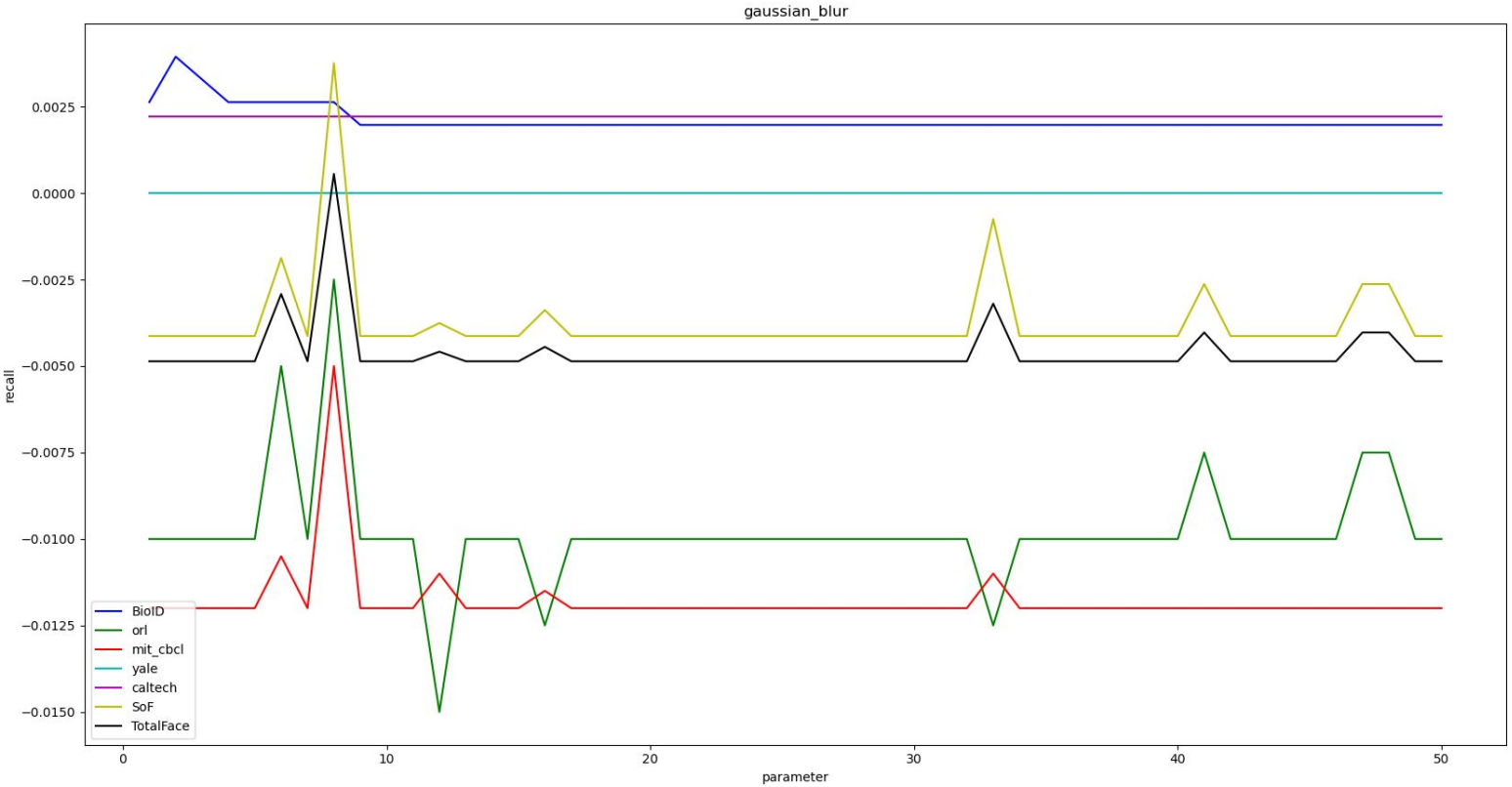
Kernel size = 5 (fixed) Sigma = 1 to 50



Sigma = 8 (seems good)

Recall graph

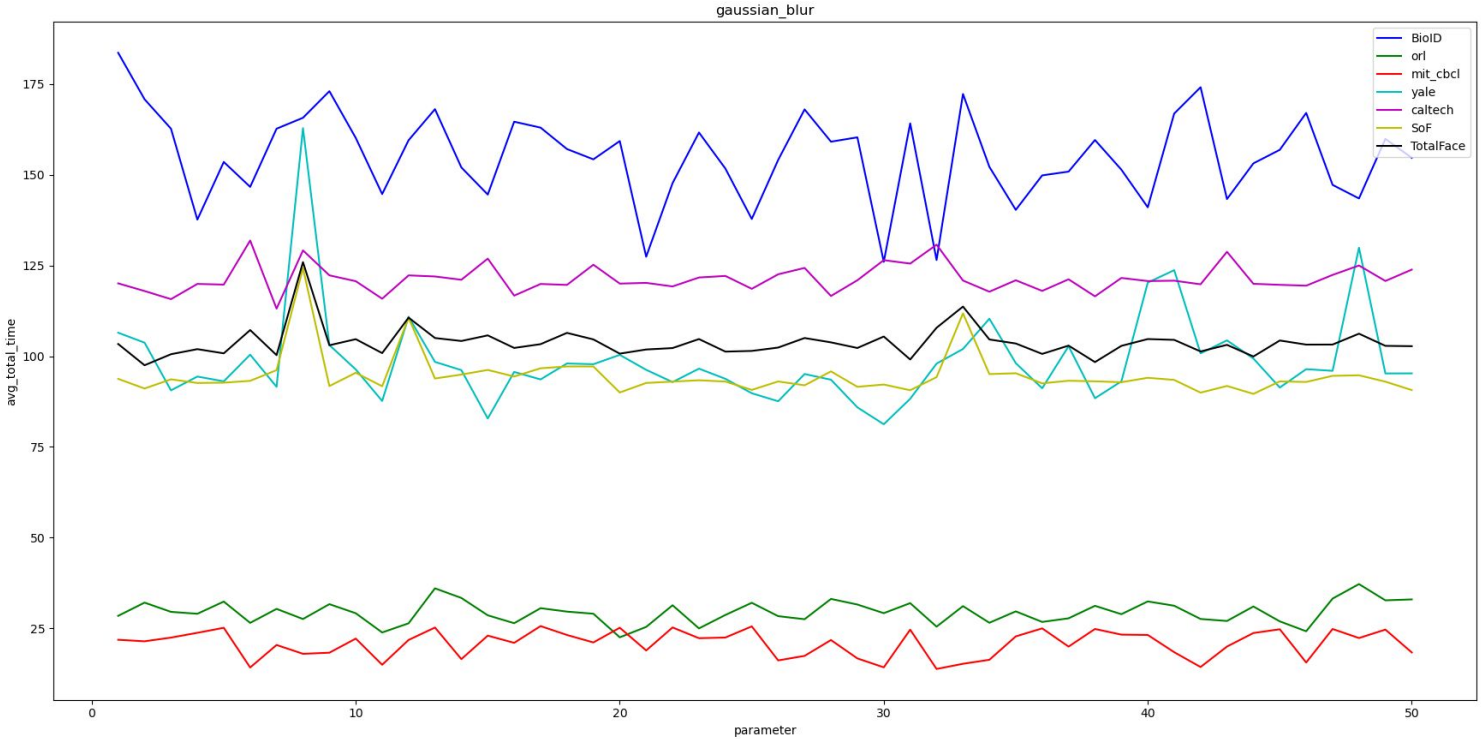
Kernel size = 5 (fixed) Sigma = 1 to 50



Sigma = 8 (seems good)

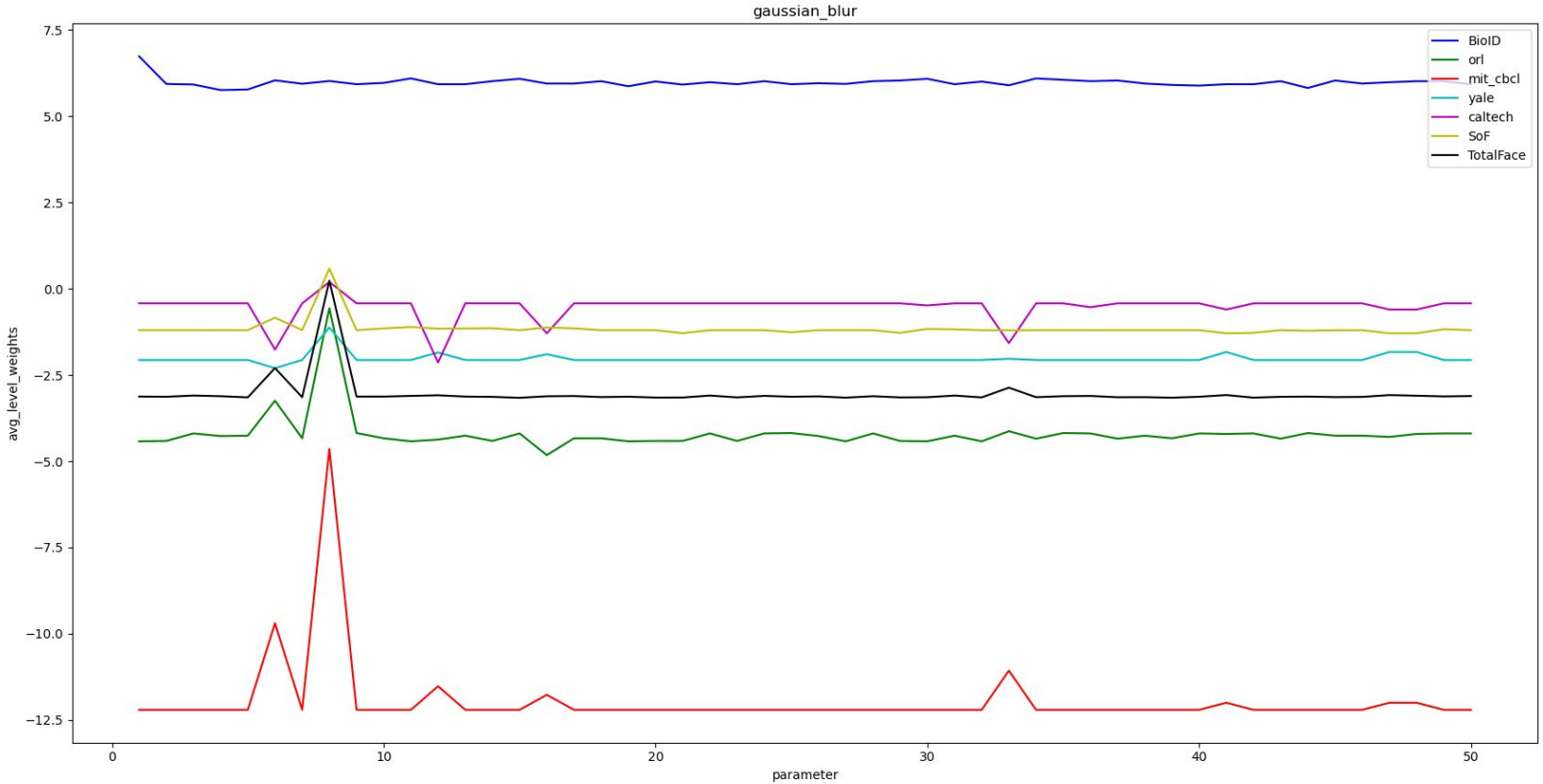
Kernel size = 5 (fixed) Sigma = 1 to 50

Time graph (percentage inc)



Kernel size = 5 (fixed) Sigma = 1 to 50

Confidence (percentage inc)

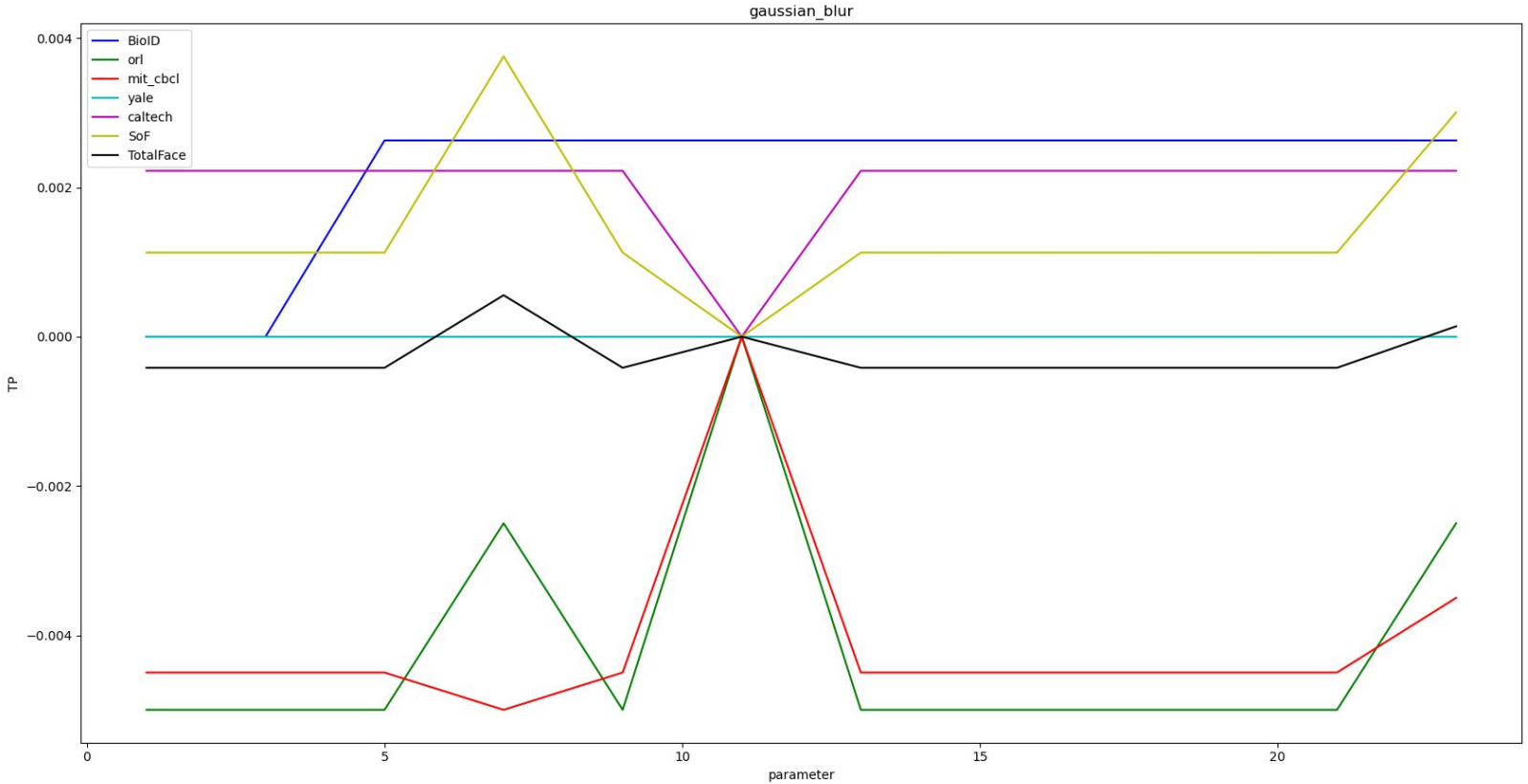


Sigma = 8 (seems good)

Sigma = 8 is apt

True Positive graph

Kernel size = 1 to 23 Sigma = 1 (fixed)



size = 7 (seems good)

To cut short:

Similar study on parameter space is done for all the pre-processings methods.

Further on, we will just take the optimal parameter of each pre-processing algorithm and then compare across algorithms.

Code for this analysis is there in codebase.

Chosen Parameters

- Gaussian Blur
  - Kernel size = 5
  - Sigma = 8
- BDA (blind deconv. algo)
  - Spatial sigma = 1
  - Intensity Sigma = auto guess scikit image
- HOMO
  - Cutoff freq. = 20
- MSR
  - Iterations =10
  - Sigma_list = 130_140_150_160_170_180_190_200_210_220
- NMBN
  - _patchSsize 10
  - H = 0.5
  - patchDistance_2
- CLAHE
  - Clip limit = 4

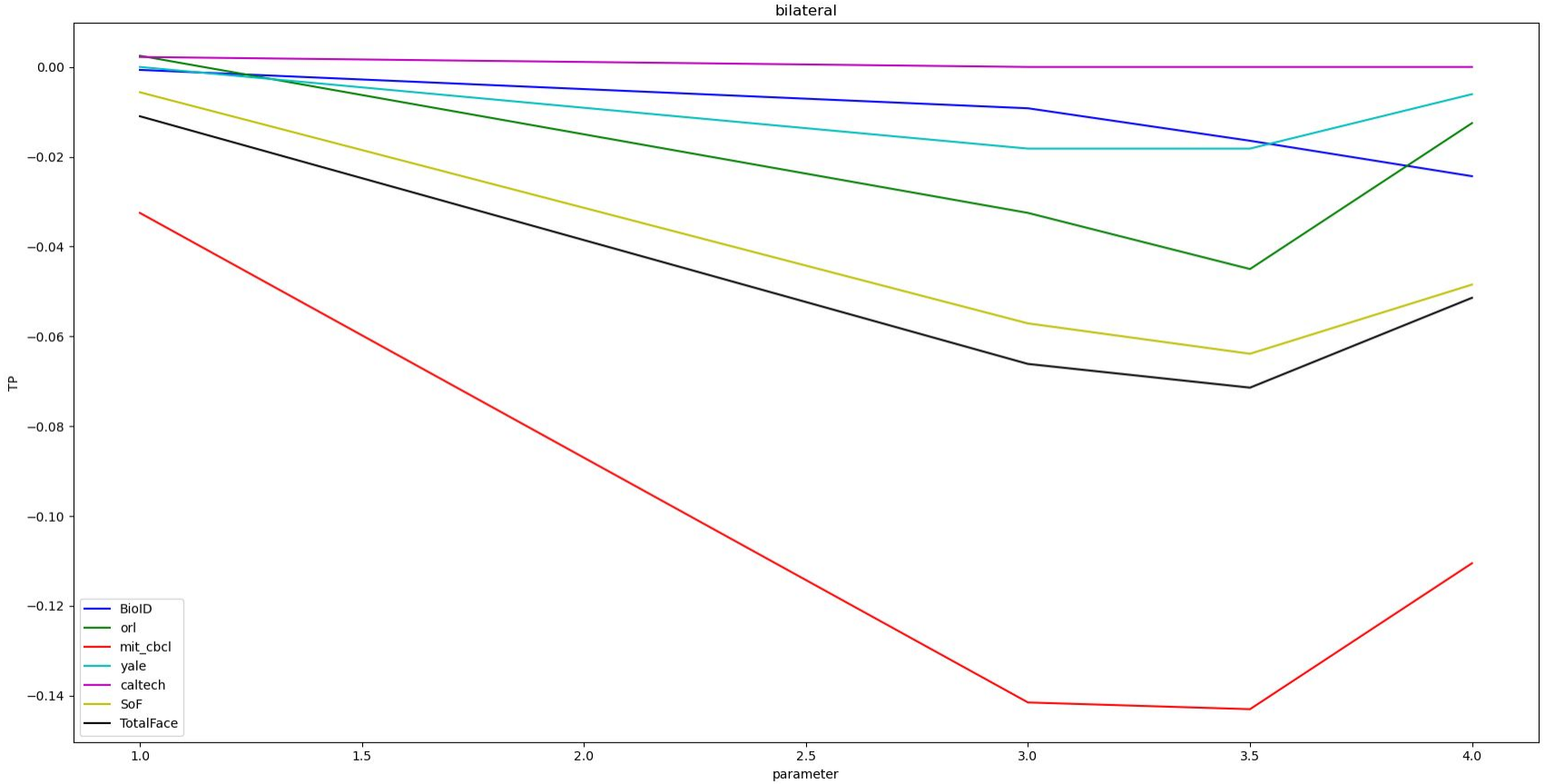
Chosen Parameters

- retinexFM
  - Iterations = 13
- SSR
  - Very bad performance
  - No parameter could save it
- TV_Chambolle
  - Weight =3

Algo without needed parameters

- Logarithmic stretch
- Full scale contrast stretch
- HE

True Positive graph Intensity sigma= auto calc (fixed) Sigma = 1 to 50



Sigma = 6 (seems good)

Table of perf comparison of all algo is in file analysis.csv

There all metrics of all methods studied

Deblurring:

Some major Issues:

- Need PSF (point spread function) for Deblurring
- Deblurring Algo's are as good as PSF Estimator
- PSF estimation in basic form has gradients and convergence (slow loops)
- SOTA PSF estimators are DNNs (!)
-
-