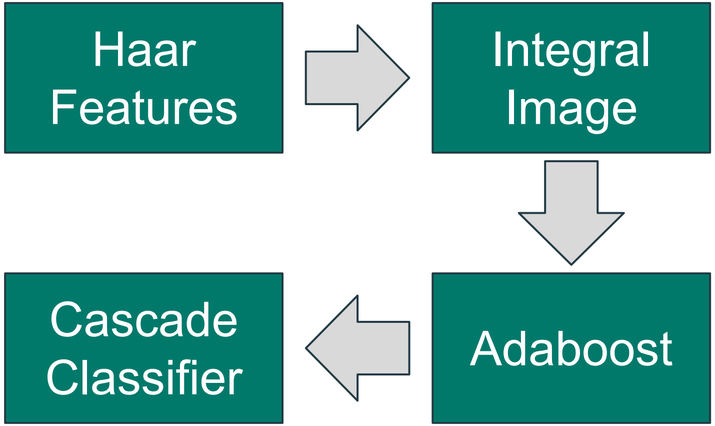
REAL TIME FACE DETECTION WITH HAAR CASCADE ROBUST TO SKIN COLOR AND VARIED ILLUMINATION

William Chandra (13215052)
Hansen (13214077)
Teddy Hadi Utama (13215086)

Viola-Jones Framework: Haar Cascade

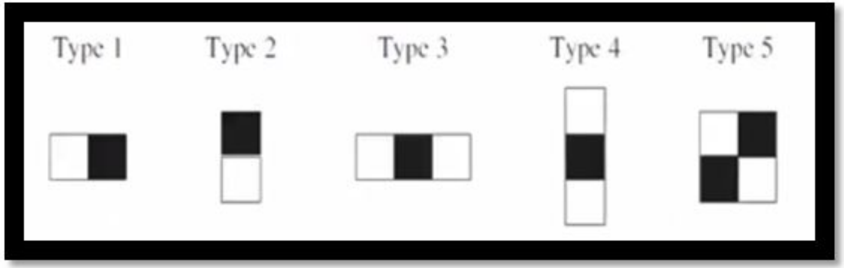
Background Theory - Viola Jones Algorithm

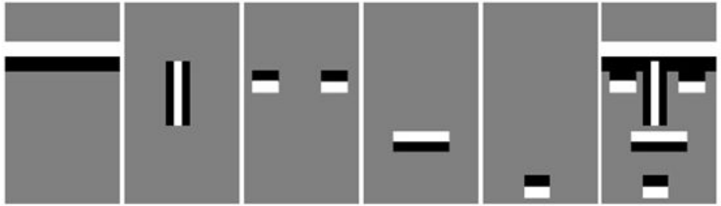


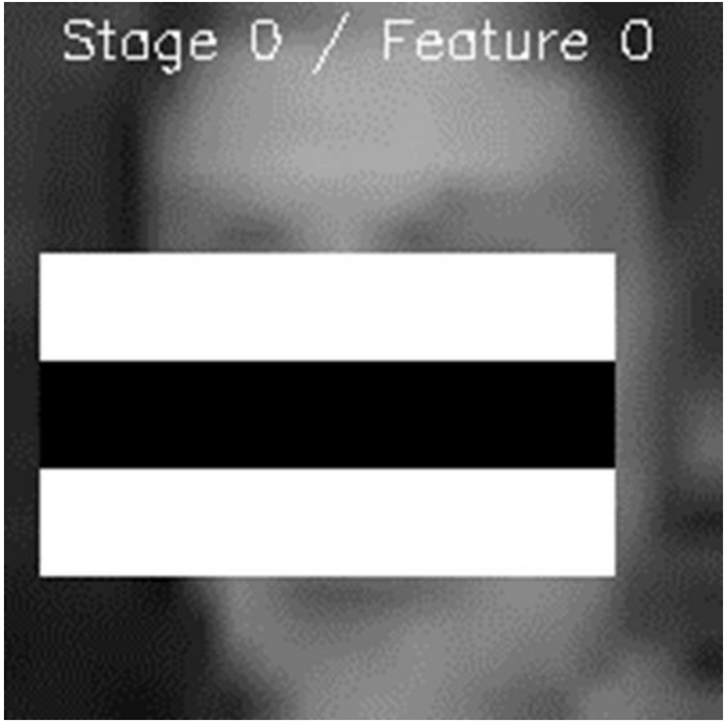
Viola Jones Algorithm **advantages** on **face detection**:

1. Fast feature computation
2. Enable feature scaling
3. Fixed computational cost
4. High detection rate and real time applicable

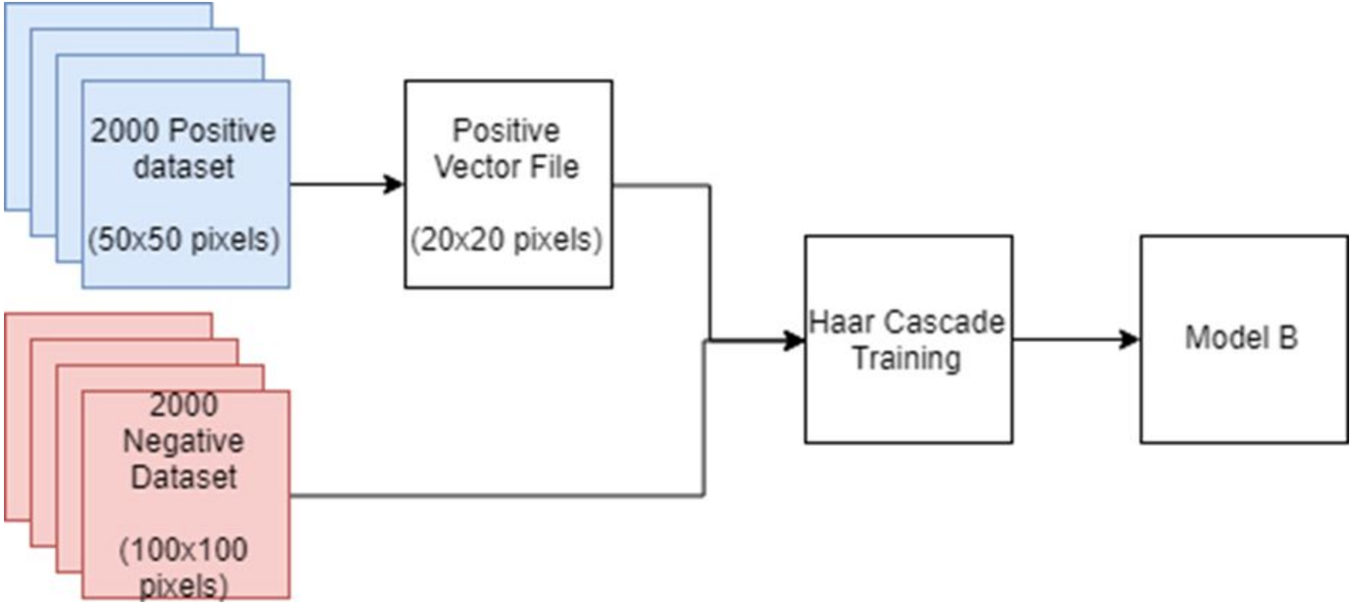
Haar Features & Integral Image







Training Custom Viola-Jones Model



Experiment:
Check the speed and accuracy of trained model

10 Multiple Faces (Total 27)

30 Single Face

39 No Face

Statistical Test

“MODEL A”
Default model from internet:
haarcascade_frontalface_default.xml[ref]

- Windows resolution: 24x24
- Number of stages: 25
- Max Number of Features/stages : 190
- Number dataset: ~10,000 images

“MODEL B”
Trained Model

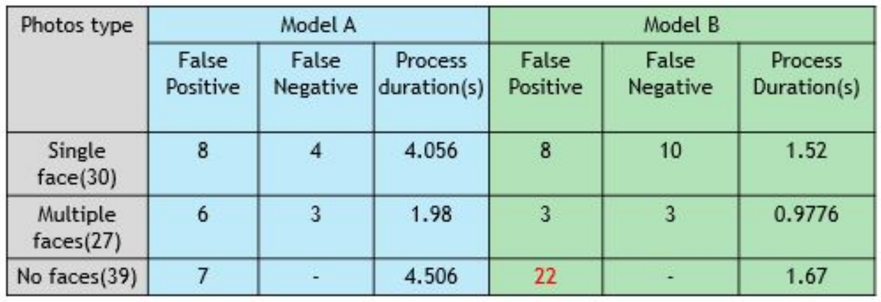
- Windows resolution: 20x20
- Number of stages: 15
- Max Number of Features/stages : 20
- Number dataset: ~2,000 images

- **False Positive** : Number of **non-face** misclassified as **face**
- **False Negative** : Number of **face** misclassified as **non-face**
- **Execution Program** : **Duration** taken to process

Here we compare two Haar Cascade model:

1. **Model A** : Default model from Internet
2. **Model B** : Trained Model

640x480 Pixels Result

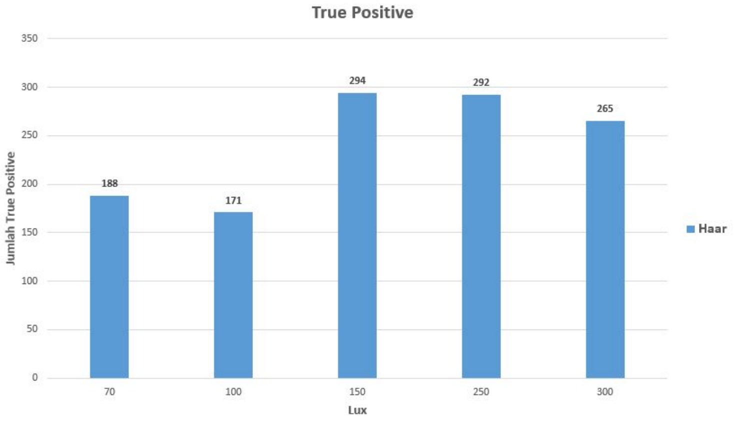


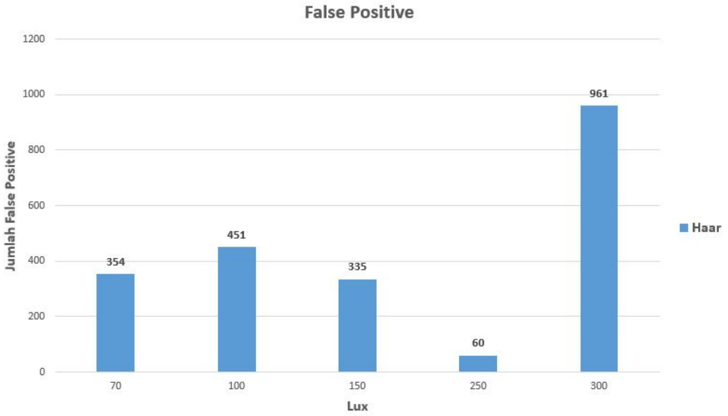
Photos type	Model A			Model B		
	False Positive	False Negative	Process duration(s)	False Positive	False Negative	Process Duration(s)
Single face(30)	8	4	4.056	8	10	1.52
Multiple faces(27)	6	3	1.98	3	3	0.9776
No faces(39)	7	-	4.506	22	-	1.67

- Model B yields <u>2,5 times faster detection process</u> than Model A
- Model B has a lot worse False Positive on negative test set than Model A
- Model B has more false negative on single pictures than Model A

Note: <u>Trained model B</u> has much <u>fast detection</u> and needed <u>to reduced</u> in <u>false positive</u>

Implementation on Larger Dataset

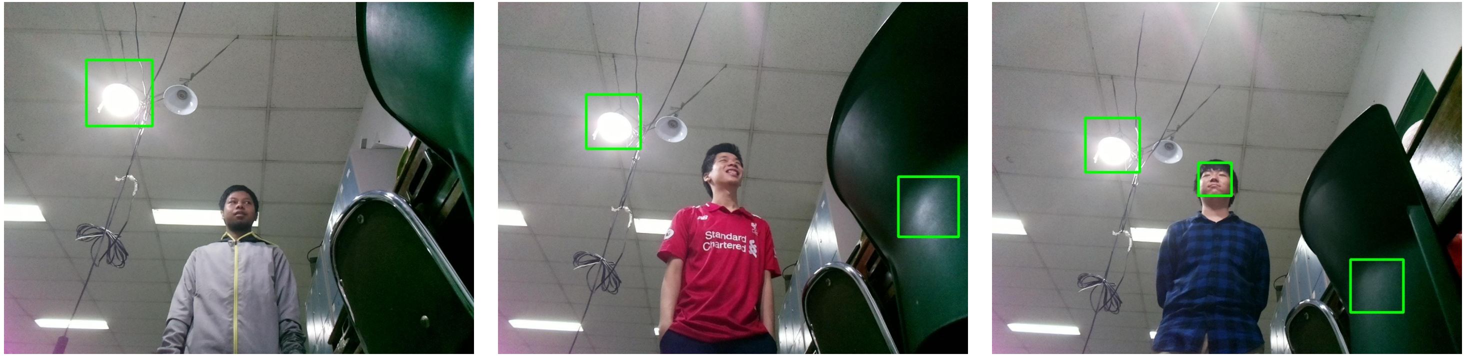




- Datasets are taken with **varieties of brightness** : 70 lux, 100 Lux, 150 Lux, 250 Lux, and 300 Lux
- 2250 Sample Images

- Measured the number of **True Positive** and **False Positive** with the varying brightness

False Positive

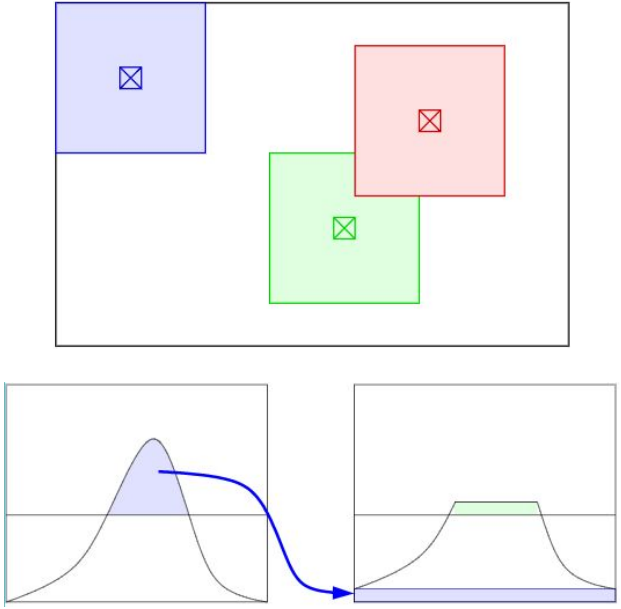


- **Illumination** problem
- Differentiating the **color** between human skin and other object

Therefore needed an algorithm that able to solve illumination problem and skin detection

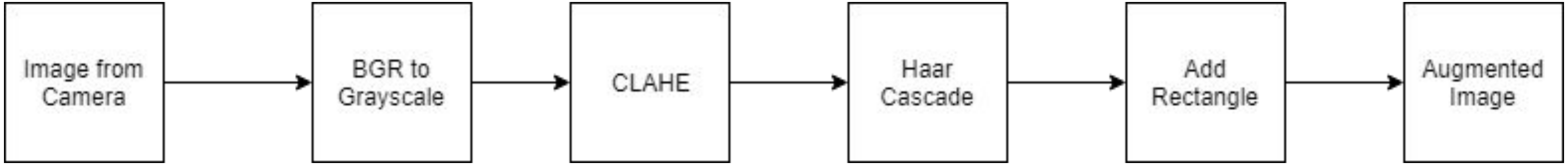
Haar Cascade with CLAHE as Preprocessing Algorithm

Background Theory - CLAHE



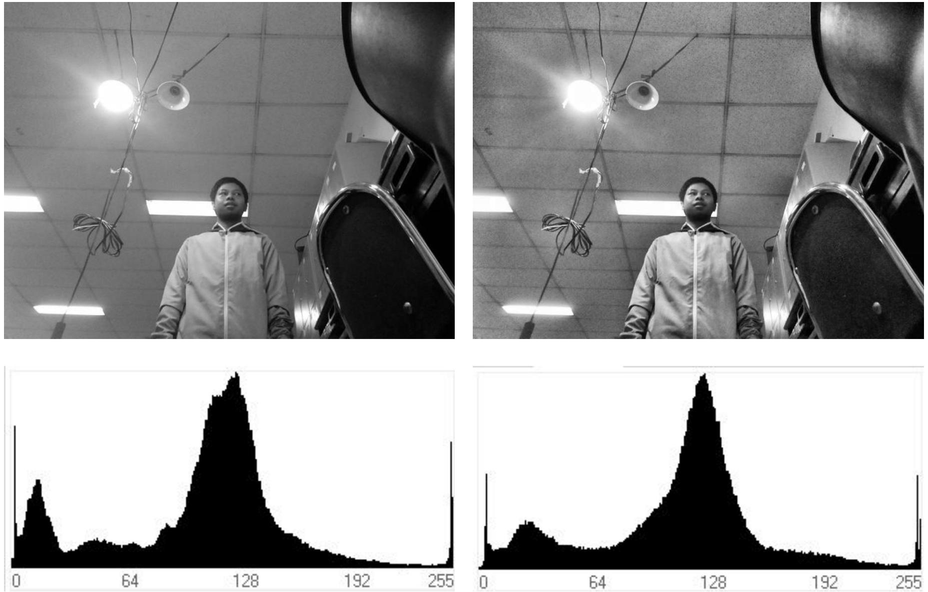
- Transforming each pixel with a transformation function derived from a neighbourhood region
- Each pixel is transformed based on the histogram of a square surrounding the pixel
- The transformation function is proportional to the cumulative distribution function (CDF) of pixel values in the neighbourhood
- CLAHE limits the amplification by clipping the histogram at a predefined value before computing the CDF

Haar Cascade Diagram with Preprocessing using CLAHE



CLAHE do **preprocessing** for every images before acting as input for Haar Cascade Algorithm

CLAHE Result on Grayscale Image

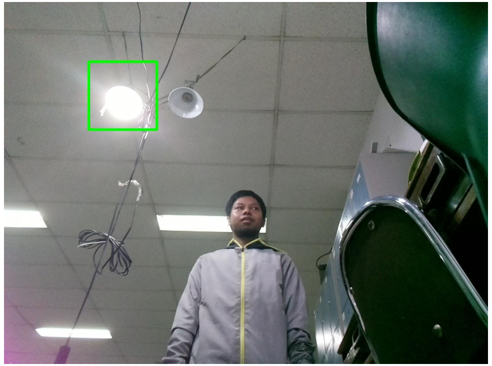

Original Image

CLAHE’d Image

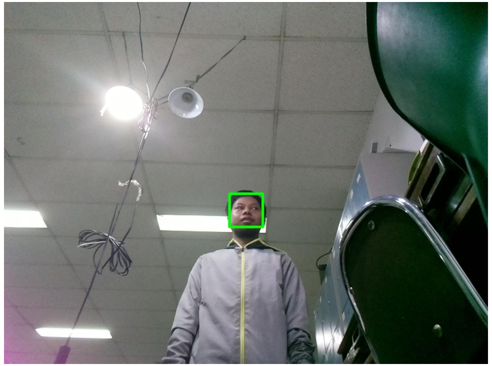
- Better image contrast
- Segmentation in image become clearer
- Reduce the wide illumination variation on the image

Face Detection Result

Haar Cascade

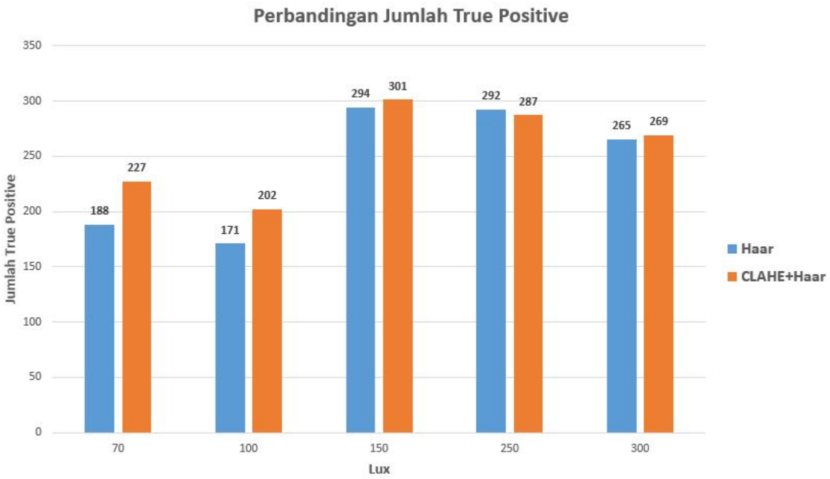


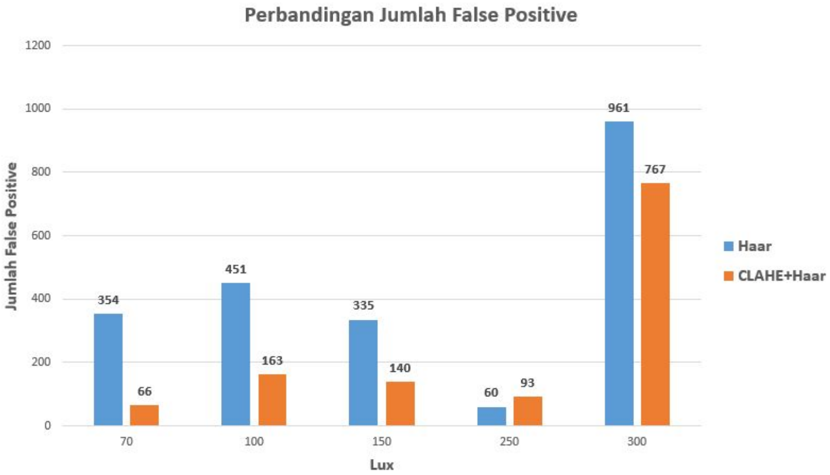
Haar Cascade + CLAHE



- Lamp no more detected as face (**reduce false positive**)
- Face is detected and **increasing** the **true positive**
- Illumination is **no more a problem** for the system

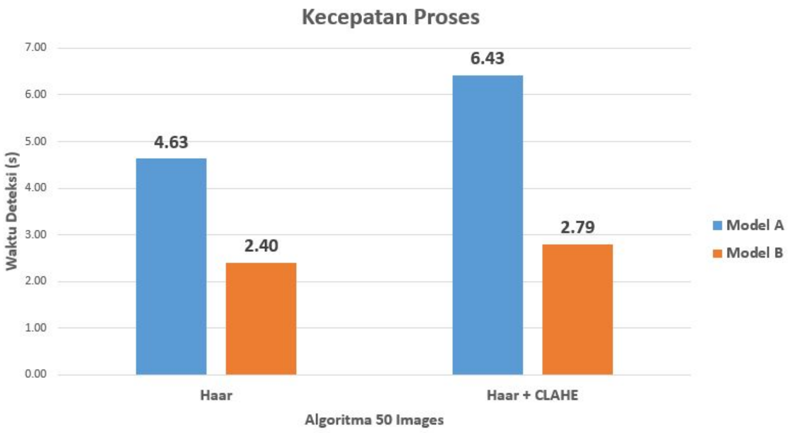
Statistical Evaluation





- Overall True Positive is increased, especially on the low brightness situation (70 lux and 100 lux)
- Overall False Positive decreased significantly

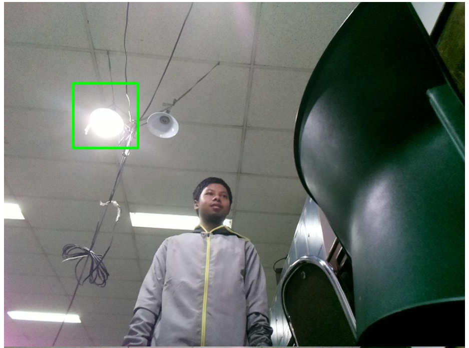
Algorithm Time Performance



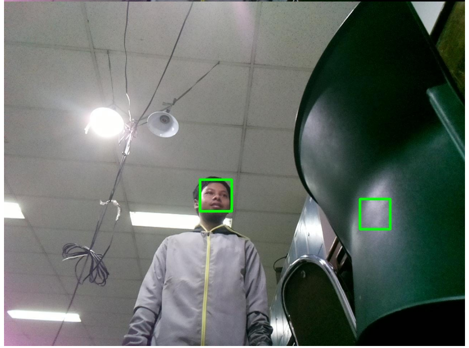
- CLAHE **didn’t affect significantly** on the overall system performance with **0.39 s** difference in processing **50 images** using **model B**
- Still **applicable** for **real time** system

Things to Improve

Haar Cascade



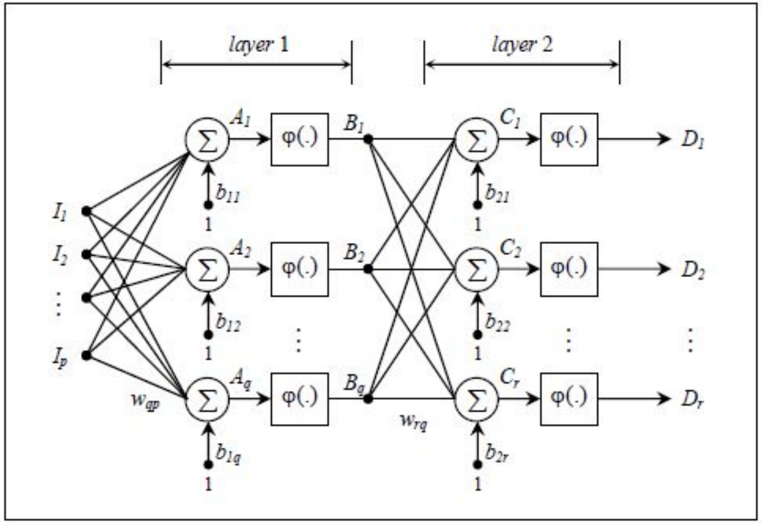
Haar Cascade + CLAHE



- CLAHE **able to reduce the illumination effect** on the images but sometimes **produce enhancement on objects that recognized as face** by Haar Cascades Algorithm
- Algorithm that able to **differentiate human skin color and non human object** is needed

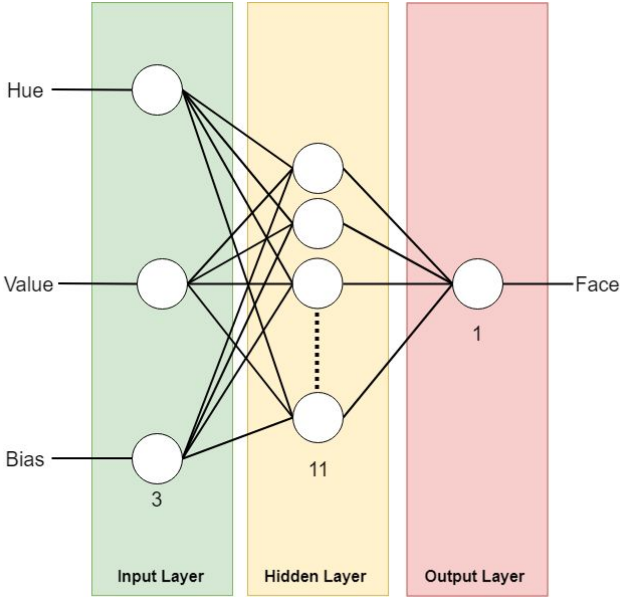
Haar Cascade Post Processing Neural Network for Face Skin Detection with CLAHE as Preprocessing Algorithm

Artificial Neural Network



- System inspired by **biological neural network**
- System **learn** the task by consider example
- Signal travel from **input layer, hidden layer and output layer**

Skin Detection (Neural Network)



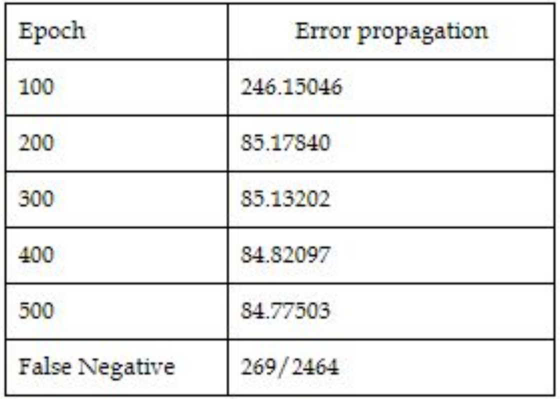
Input parameter Skin Detection :

- Hue
- Value

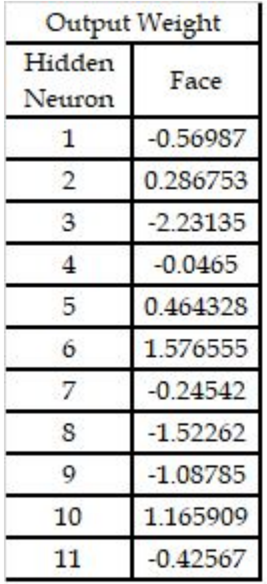
Output parameter Skin Detection:

- Face/No-Face

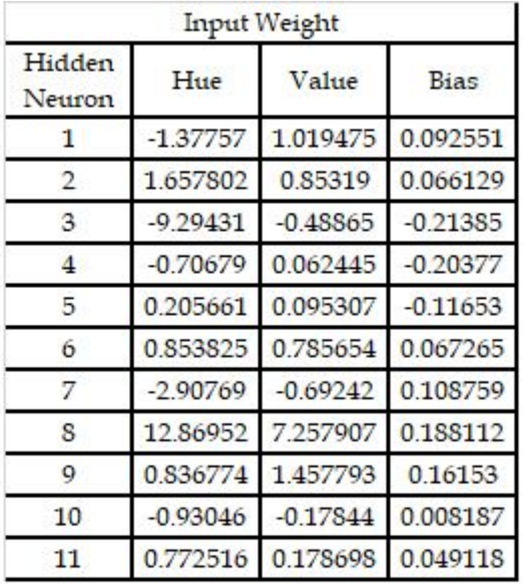
Back Propagation



Epoch	Error propagation
100	246.15046
200	85.17840
300	85.13202
400	84.82097
500	84.77503
False Negative	269/2464

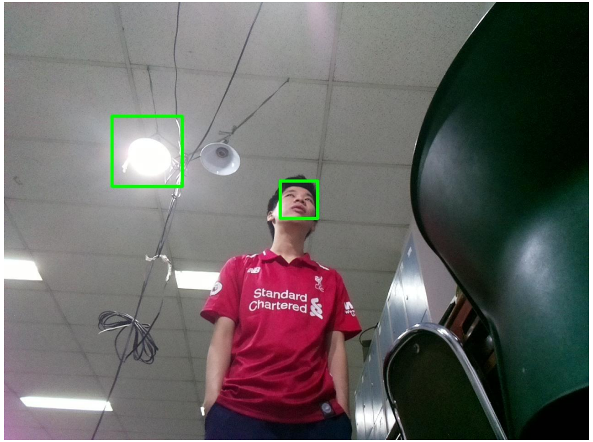


Output Weight	
Hidden Neuron	Face
1	-0.56987
2	0.286753
3	-2.23135
4	-0.0465
5	0.464328
6	1.576555
7	-0.24542
8	-1.52262
9	-1.08785
10	1.165909
11	-0.42567

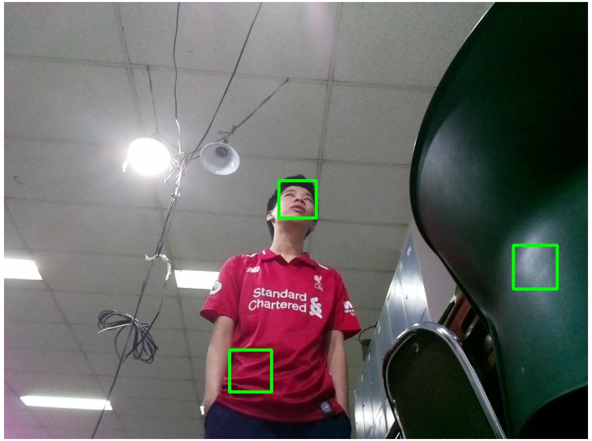


Input Weight			
Hidden Neuron	Hue	Value	Bias
1	-1.37757	1.019475	0.092551
2	1.657802	0.85319	0.066129
3	-9.29431	-0.48865	-0.21385
4	-0.70679	0.062445	-0.20377
5	0.205661	0.095307	-0.11653
6	0.853825	0.785654	0.067265
7	-2.90769	-0.69242	0.108759
8	12.86952	7.257907	0.188112
9	0.836774	1.457793	0.16153
10	-0.93046	-0.17844	0.008187
11	0.772516	0.178698	0.049118

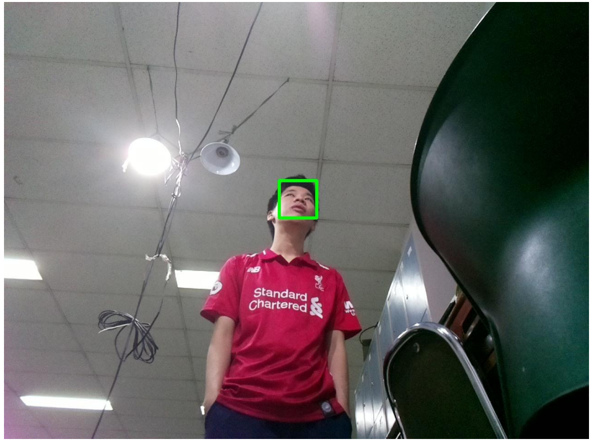
Fixing Image Using Neural Network



Haar Cascade

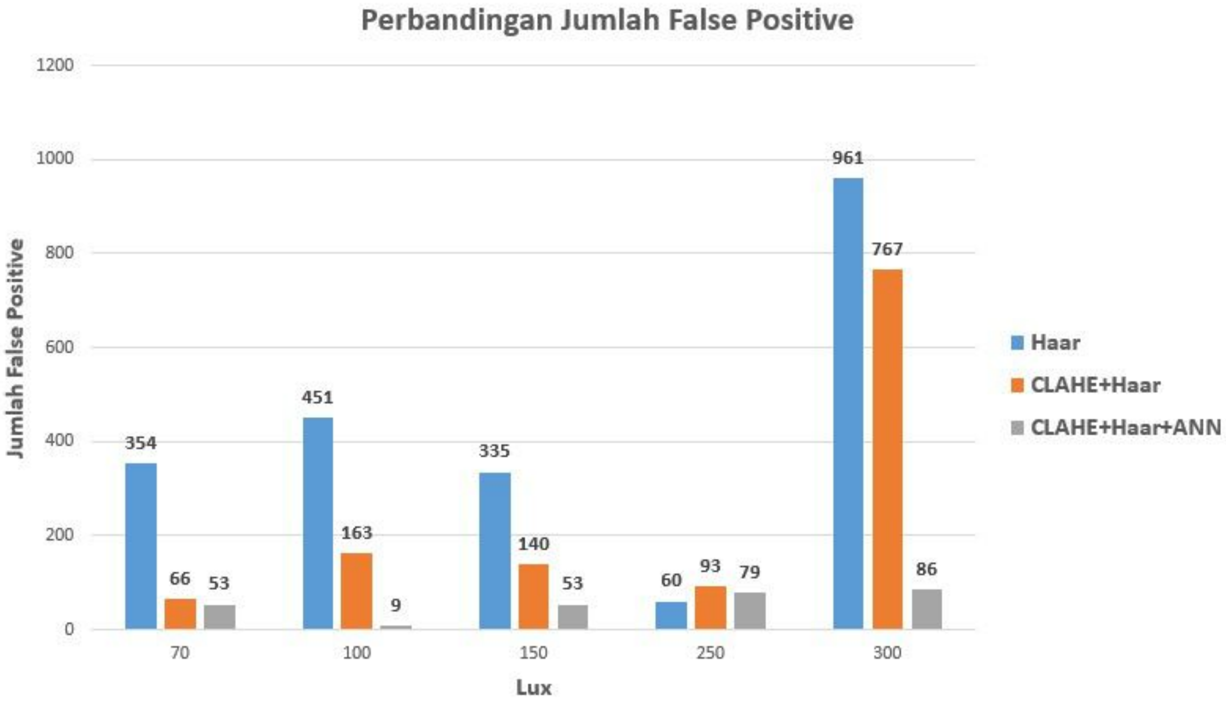


Haar Cascade + CLAHE

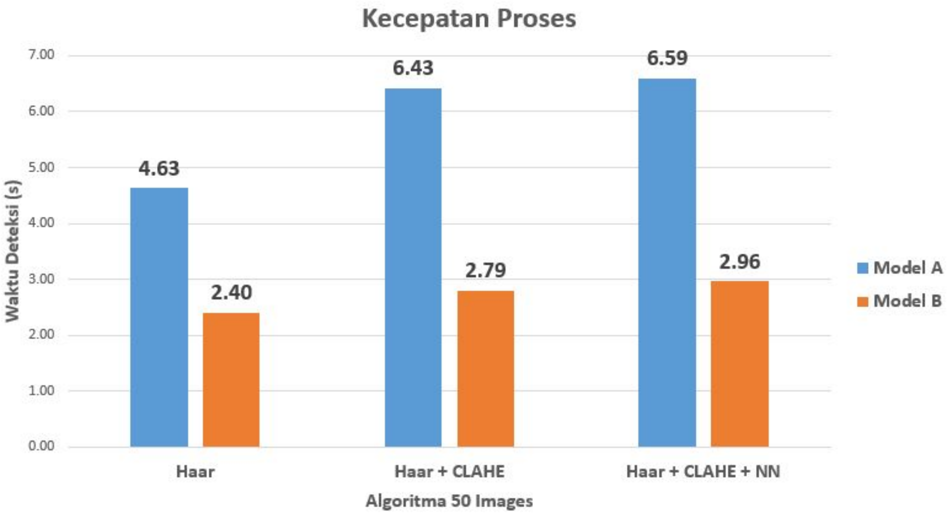


Haar Cascade + CLAHE + NN

Comparison: Algorithm False Positive



Comparison: Algorithm Performance Time



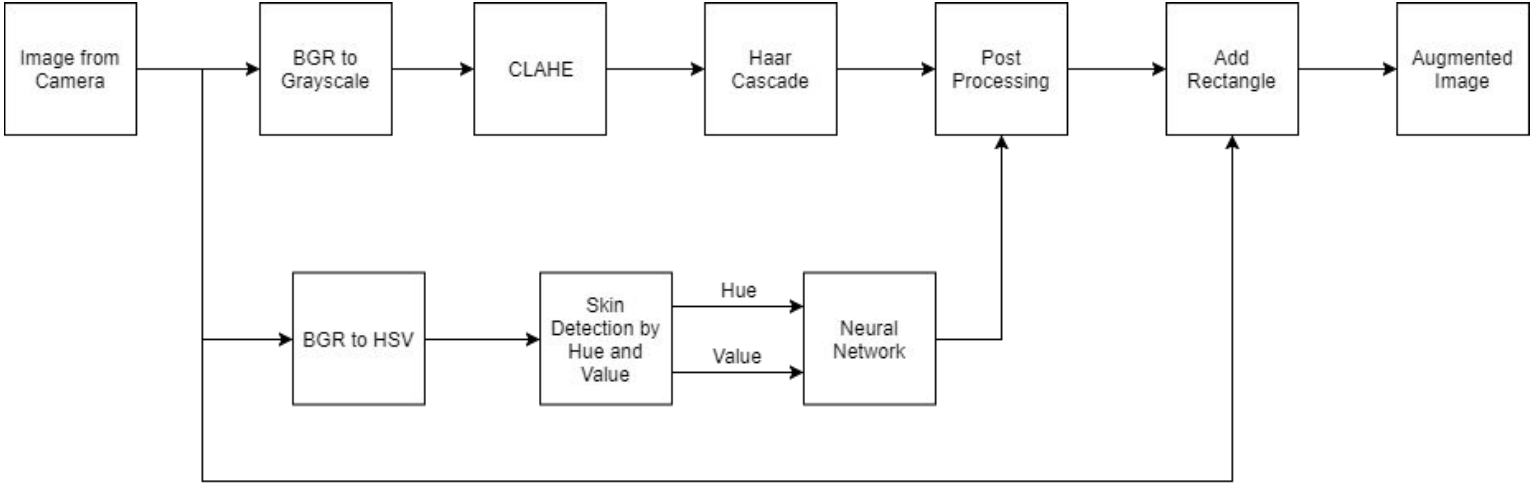
On model B:

Haar + CLAHE + NN only **0.56s slower** than Haar Cascade alone in processing 50 images.

Thus still **applicable for real time face detection**

Real Time Implementation

System Block Diagram



Live Detection

- Laptop
- Camera
- Python with OpenCV library

DEMO

Questions?

THANK YOU