ETAPA 1 – VISÃO NOITE

Gabriel Pinho

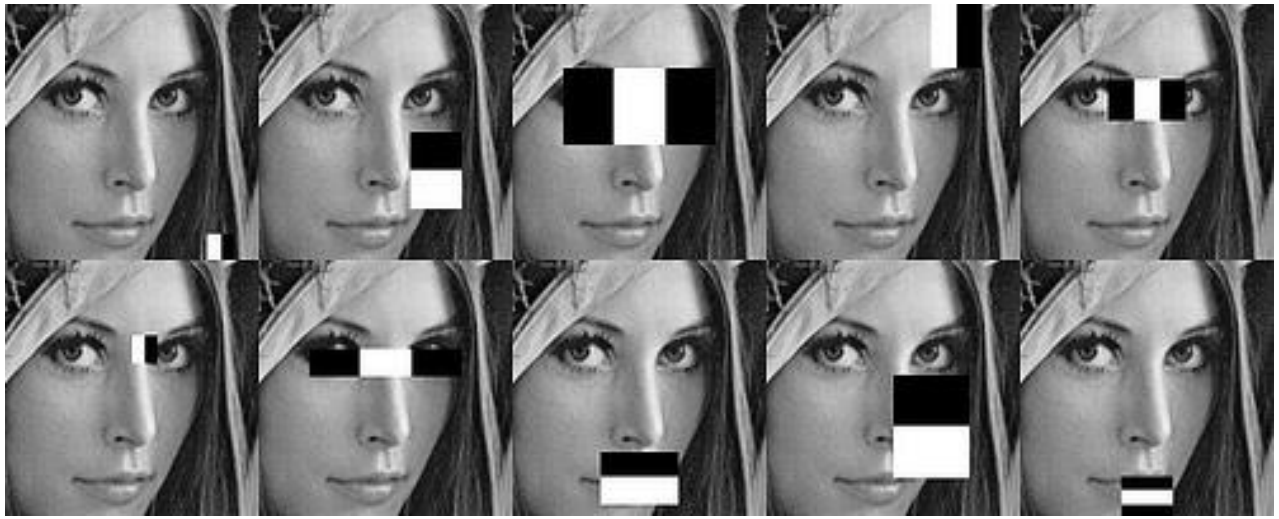
Jonathan paixão

Matheus Bezerra

Samuel Petroline

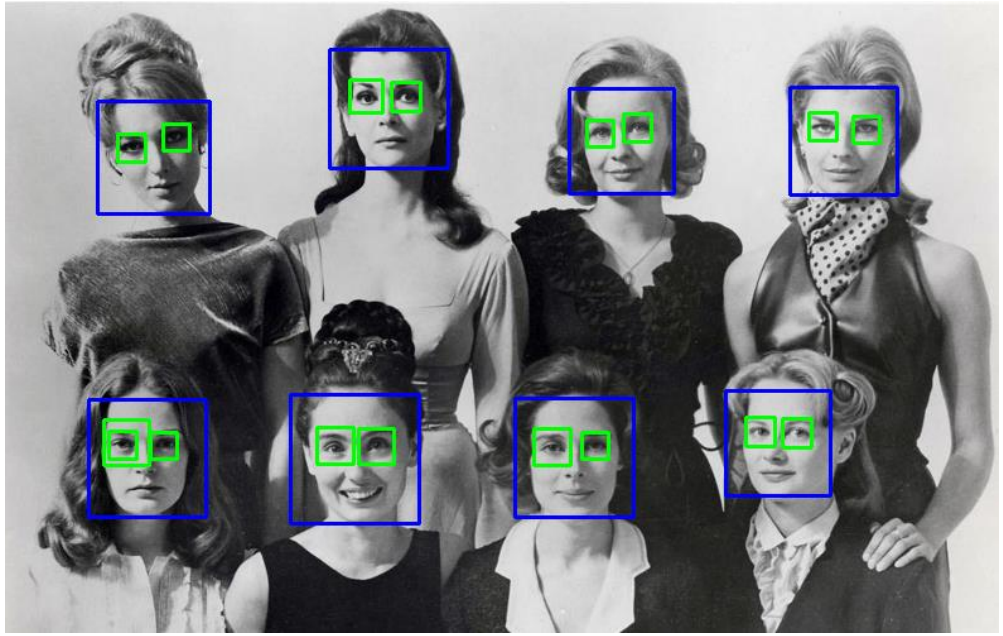
Algoritmo Haar Cascade

- Esse algoritmo utiliza máscaras para caracterizar um objeto por meio de variações de luminosidade. As máscaras capturam essas variações em diferentes amplitudes e direções, e os valores que caracterizam um certo tipo de objeto são aprendidas com um algoritmo de aprendizagem de máquina para gerar vários classificadores



Algoritmo Haar Cascade

- Uma vez que esses classificadores são produzidos isto é, treinados a partir de imagens de exemplo, ele encontra o objeto em uma nova imagem executando os vários mini classificadores "em cascata"



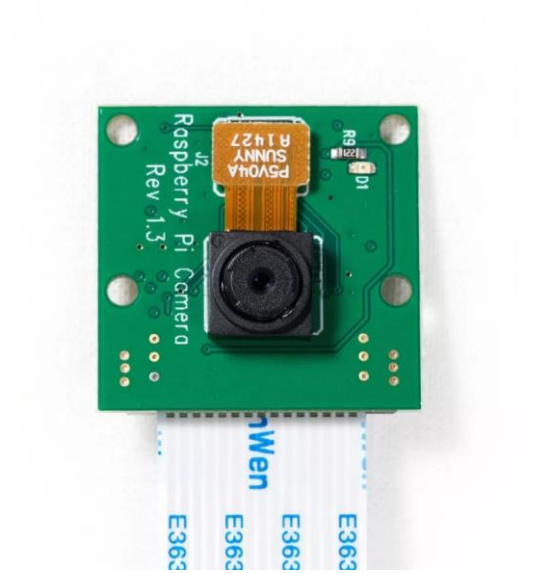
Identificação de rostos de Peixes

- Evitar epidemias de piolhos do mar – parasitas marinhos que podem dar prejuízos de bilhões para criadores de salmão
- As câmeras são capazes de fazer uma varredura tridimensional das cabeças dos peixes e identificá-los por meio de elementos que os tornem únicos
- Os peixes que estiverem com piolhos do mar serão prontamente retirados do convívio com os outros e tratados devidamente.

" Desafio dos 10 anos"

- Desafio dos 10 anos, em que as pessoas devem compartilhar uma foto atual ao lado de uma imagem de 2009, para fazer um comparativo.
- Kate O'Neill, fundadora da KO Insights e escritora do livro Tech Humanist.
- Segundo O'Neill, esse tipo de sistema pode utilizar as imagens não apenas para identificar o rosto do usuário, mas também para aprender sobre progressão de idade e características ligadas ao envelhecimento.

Modulo Camera P/ Raspberry Pi 5mp + Cabo Flat



É possível usar uma câmera USB

- Resolução para imagens: 2592x1944
- Vídeo:
  - 1080p 30fps
  - 720p 60fps
  - 640x480 90fps
- Campo de Visão: 2,0 x 1,33 m, a 2 m

- R$ 50,00 + frete

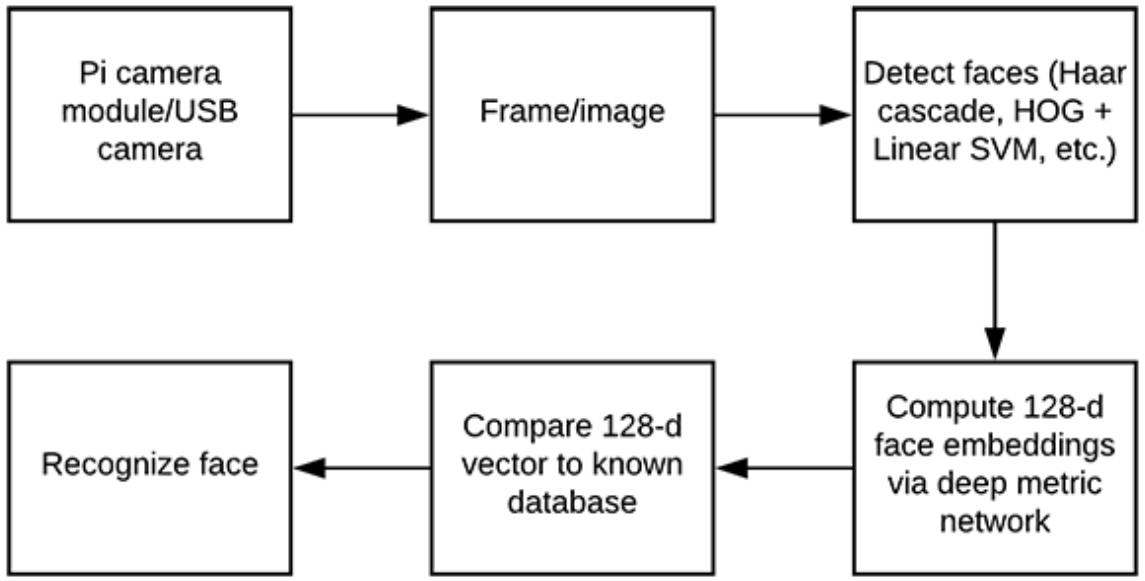
https://github.com/jrosebr1/imutils

"A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and ***both*** Python 2.7 and Python 3."

Micro SD 16 GB Sandisk Class 10

- 16GB
- Class 10
- Até 40 Mb/s de leitura

- R$ 21,24 + frete (Americanas)



Considerações

- Processo complexo para se realizar em um Raspberry Pi
- Quanto mais pessoas (imagens) para reconhecer, mais devagar o algoritmo
- Com 3 pessoas: 1 ~ 2 FPS
- Com otimização, no máximo 8 FPS

Como melhorar:

- Ao invés de reconhecer com captura de vídeo em tempo real, tira uma foto e faz o reconhecimento da foto.

Referência: https://www.pyimagesearch.com/2018/06/25/raspberry-pi-face-recognition/

Reconhecimento facial

Utilizando OpenCV
+ Python

PASSOS

1o. Passo - Encontrando o rosto:

É utilizado um algoritmo chamado **Histogram of Oriented Gradients (HOG)**

2o. Passo – Centralizando as faces

Deforma e centraliza os rostos para que haja padronização no reconhecimento. É utilizado um algoritmo chamado **face landmark estimation**

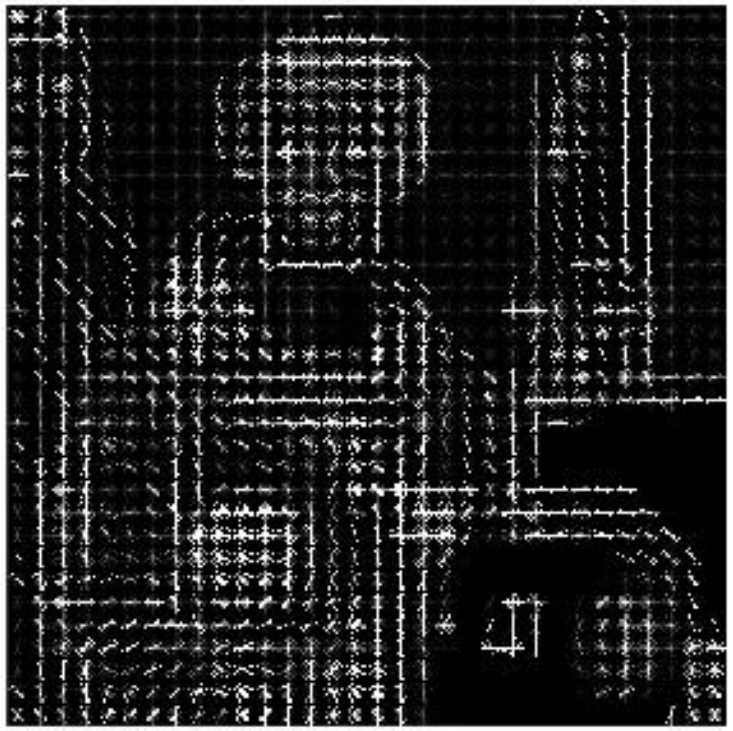
3o. Passo – Analisando e codificando

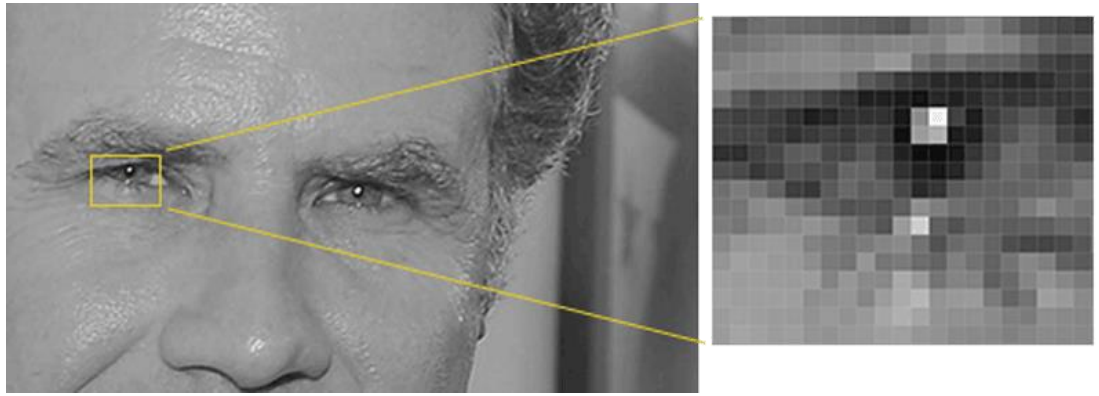
Extrai medidas dos rostos (como o tamanho da orelha, espaçamento entre os olhos, etc) que irão alimentar uma rede neural (Deep Convolutional Neural Network) que será responsável por codificar cada rosto

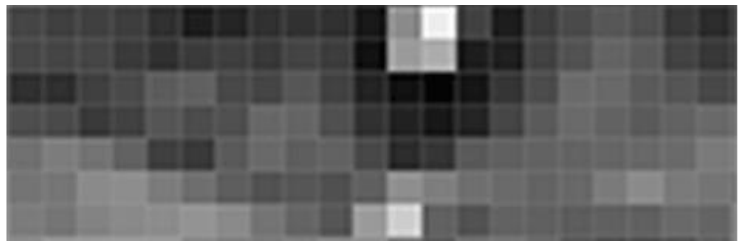
1o. Passo: encontrando os rostos utilizando HOG

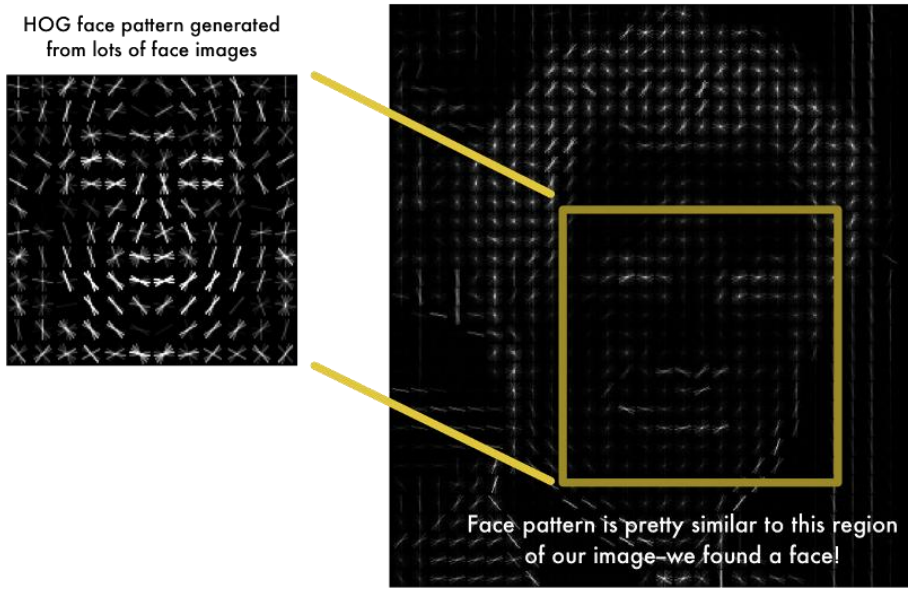
Input image

Histogram of Oriented Gradients



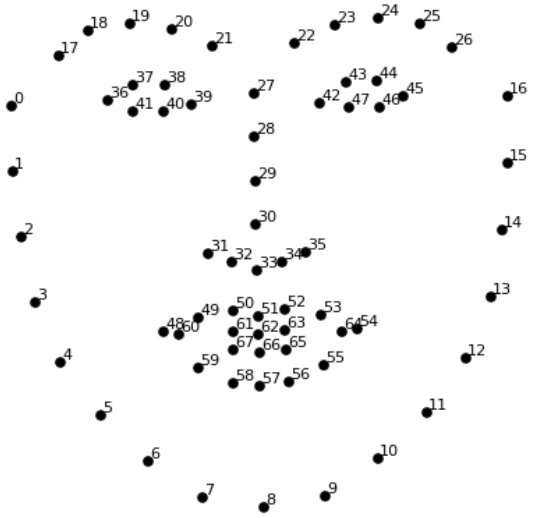




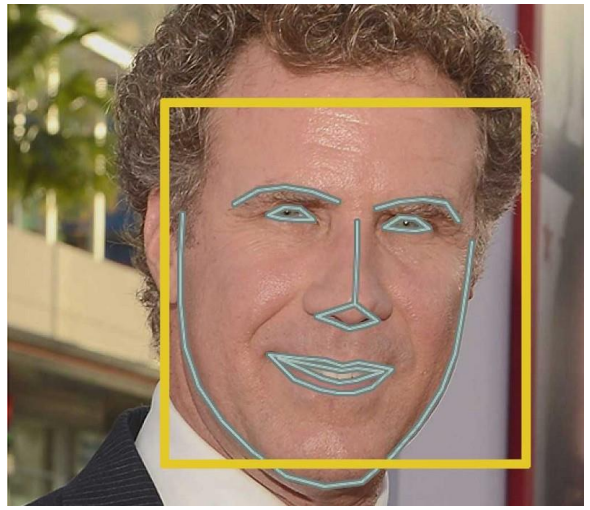


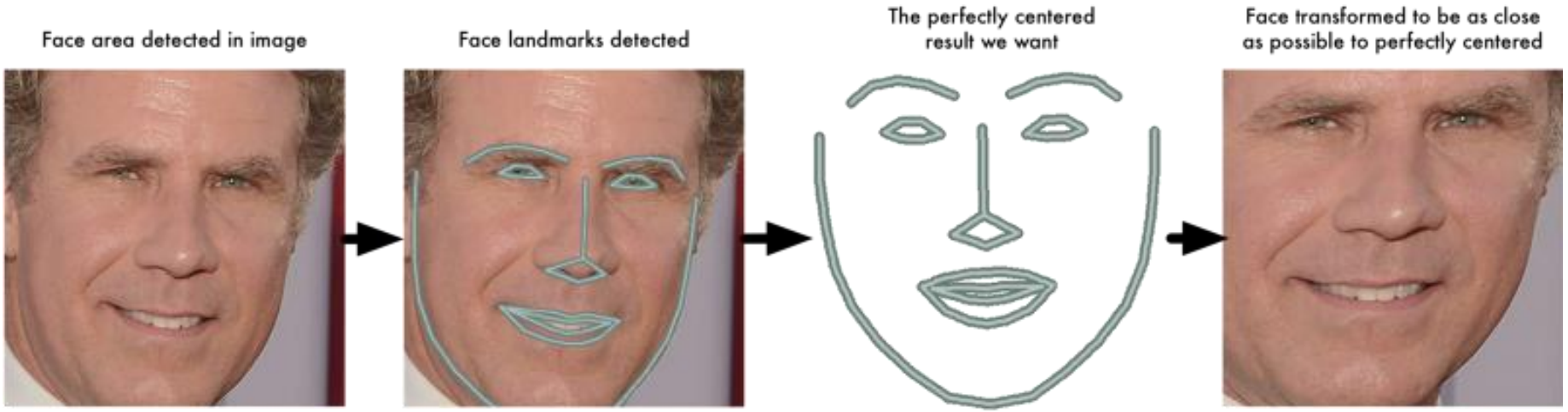
Divide pixels em blocos (geralmente 8x8)

2o. Passo: centralizando as faces

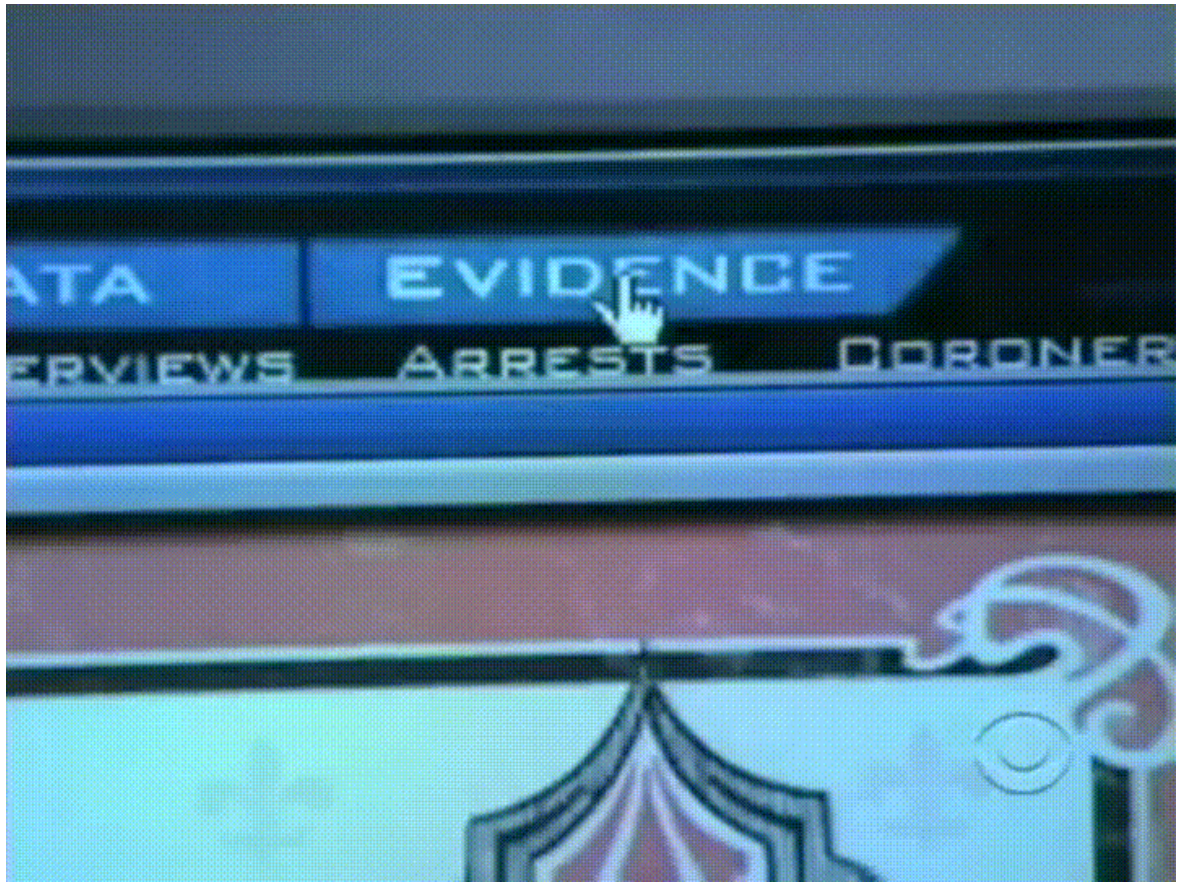


Landmarks

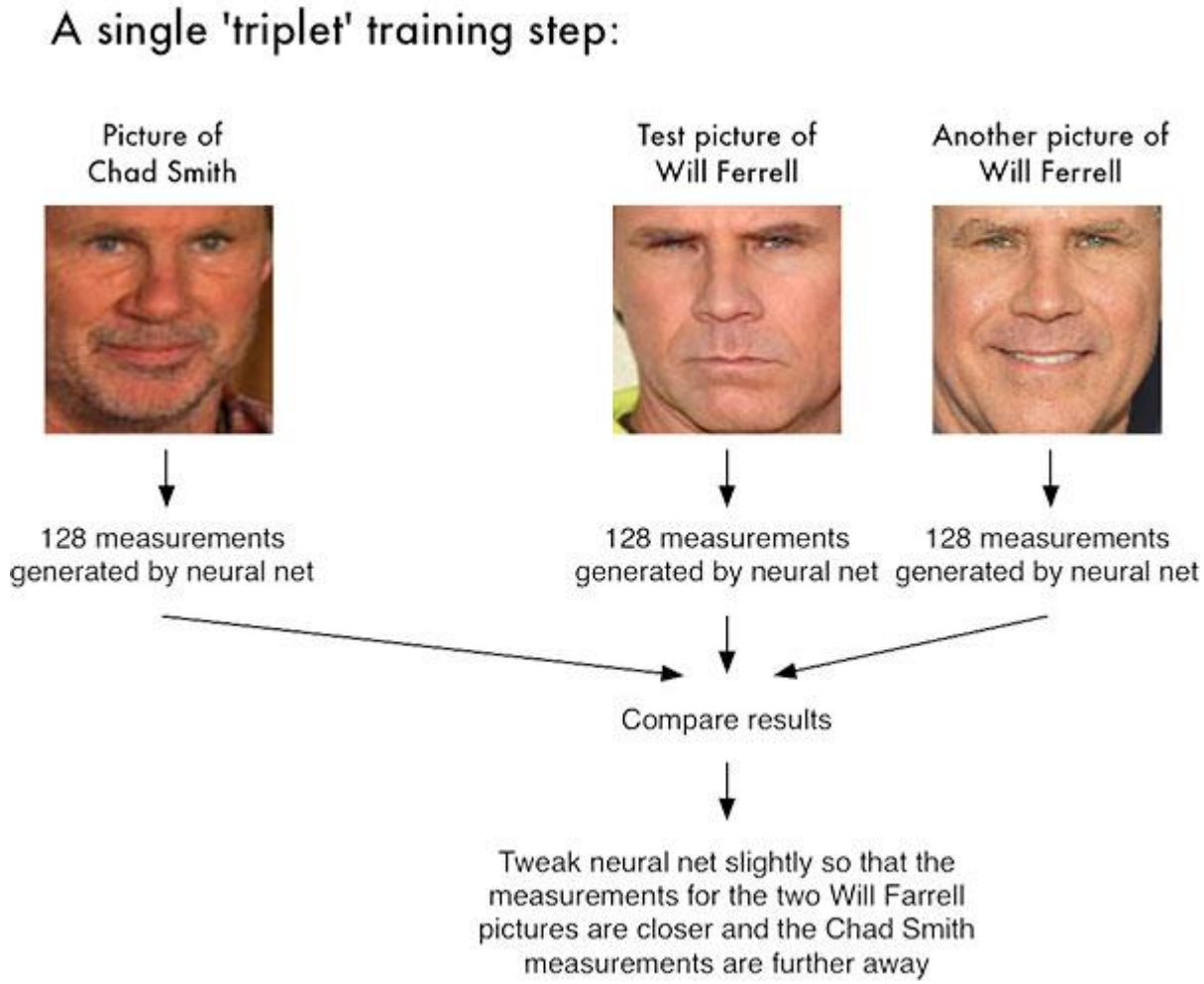




3o. Passo – Analisando e codificando



Treina uma rede neural para determinar quais medidas deve se extrair de cada imagem



Já existem modelos de machine learning prontos para extração das medidas

Ao todo, para cada imagem são geradas 128 medidas (que ninguém sabe quais são)

Input Image

128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

4o. Passo – Reconhecendo as pessoas

Utiliza-se um algoritmo simples de classificação para comparar um novo input com a base de dados gerada no passo

1 – Raspbain

- Raspbian é ideal para quem está iniciando no Raspbarry PI.
- “sistema operacional padrão”
- permite navegar na Internet e possui uma grande quantidade de ferramentas de configuração do computador e de desenvolvimento.

2 - Ubuntu

- O Ubuntu é a distribuição Linux
- Trata-se de uma edição mais leve e que roda de forma satisfatória nos Raspberry Pi 2 e Raspeberry Pi 3
- Ubuntu Mate é um sistema para quem quer fazer do Raspberry um desktop mais tradicional.

3 - Pidora

- O nome esquisito refere-se à junção de Pi com Fedora, nome da distribuição Linux da Red Hat
- Um recurso bem interessante do Pidora é o chamado modo Headless
- pode significar um modo facilitado de usar o Raspberry na falta de um monitor dedicado.

4 - PiPplware

- O PiPplware é uma distribuição Linux portuguesa para o Raspberry Pi
- Esta distribuição foi criada e desenvolvida por Diogo Santos
- , não sendo necessário muitos conhecimentos técnicos de Linux

REFERÊNCIAS

- https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/
- https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
- https://github.com/ageitgey/face_recognition