Michel André Lima Vinagreiro

TUTORIAL FINAL SOBRE O ALGORITMO UTILIZADO PARA PREDIÇÃO INTEGRANTE DO PROJETO DE RECONHECIMENTO FACIAL DE APLICAÇÃO VEICULAR

Faculdade de Tecnologia de Santo André

Janeiro de 2020

O objetivo deste tutorial é permitir a reprodução dos experimentos, pesquisa e resultados por meio da explicação da concepção do projeto, métodos utilizados, ferramentas confeccionadas e aplicadas e efeitos observados.

1. Considerações iniciais.

Antes de iniciar a reprodução deste segundo ponto da pesquisa é importante que todas as etapas referentes à obtenção da base de dados e o treinamento da rede CNN sejam devidamente realizadas.

1.2. Ferramentas de *hardware* utilizadas.

Na etapa de predição o *hardware* utilizado basicamente consiste na plataforma embarcada denominada *RaspBerry Model 3B* acrescido dos periféricos básicos utilizados em computadores pessoais e uma câmera específica desenvolvida especialmente para a arquitetura. Os circuitos de acionamento via relé são considerados dispositivos a parte.

1.3. Ferramentas de *software* utilizadas.

O primeiro passo consiste em instalar a interface de aplicações denominada *Python*, em sua versão *3.6*. A instalação, no sistema operacional, é realizada utilizando o comando:

sudo apt-install python 3.6.8

Por padrão, o instalador de pacotes do *python*, *pip*, é instalado como componente durante o processo acima descrito.

A partir do ponto em que a instalação foi concluída, o *pip* é utilizado para realizar a instalação das bibliotecas necessárias.

Uma vez que os procedimentos de instalações de pacotes utilizando o instalador *pip* já foram explicados no tutorial de treinamento, abaixo apenas serão informados os pacotes que devem ser instalados no sistema:

Tabela 1 - Pacotes instalados no Raspberry Pi.

Pacote	Versão	Pacote	Versão
absl-py	0.8.1	numpy	1.17.3
astor	0.8.0	opencv-contrib-python	3.4.3.18
Click	7.0	Pillow	6.2.1
dlib	19.18.0	pip	10.0.1
face-recognition	1.2.3	protobuf	3.10.0
face-recognition-models	0.3.0	PyYAML	5.1.2
gast	0.3.2	scikit-learn	0.21.3
gcc	0.1	scipy	1.3.1
google-pasta	0.1.7	setuptools	39.0.1
grpcio	1.24.3	six	1.12.0
h5py	2.10.0	tensorboard	1.13.1
joblib	0.14.0	tensorflow	1.13.1
Keras	2.3.1	tensorflow-estimator	1.14.0
Keras-applications	1.0.8	termcolor	1.1.0

Keras-Preprocessing	1.1.0	werkzeug	0.16.0
Markdown	3.1.1	whell	0.33.6
Wrapt	1.11.2		

2. Etapas de predição e a aplicação *identvideo_gpio.py*

2.1. Carregando os arquivos anteriormente treinados.

Após o termino dos processos de treinamento e geração de todos os arquivos necessários para predição, os primeiros passos consistem em carregar para dentro do programa, os arquivos anteriormente treinados (base de dados e a rede CNN).

As linhas de programa contendo os nomes e localização dos arquivos da base de dados (extensão .pickle) e os arquivos contendo a arquitetura treinada da rede CNN (extensão .h5) devem ser editadas de acordo com o nome e o caminho da pasta onde esses arquivos se encontram.

O programa atribui a imagem de uma face arbitraria como sendo de um determinado usuário se a face for pré-classificada na etapa de localização de face, se a face for classificada como pertencente ao usuário a qual foi pré-classificada e se a distancia euclidiana em relação a face padrão deste usuário for menor que um limiar anteriormente estabelecido.

Além dos arquivos de base de dados e arquitetura CNN, o programa deve carregar inicialmente as faces padrões dos usuários cadastrados, exceto para o usuário “intruso”.

Na parte inicial do programa são definidas também as configurações do *hardware* físico dedicado, tais como as configurações de *GPIO*, nesse momento, definindo qual pino será utilizado como acionamento da interface relé. A saída utilizada foi a de número 12 da placa.

2.2. Captura instantânea de *frame* através da câmera.

Após o programa ativar a câmera, a entrada de caracteres via teclado é ativada e um *frame* é capturado e submetido a um localizador de face baseado no algoritmo *HOG*. Após determinar a região onde se localiza a face e estabelecer seus limites, o vetor de características da face é extraído e comparado como os vetores pertencentes a base de dados, se a similaridade for maior para os vetores classificados como “intrusos” o programa rapidamente captura outro *frame*, se a maior similaridade for relacionada a vetores atribuídos a um determinado usuário o programa prossegue e realiza verificações adicionais.

2.3. *Verificações realizadas para cada usuário*.

Após a face recortada ser pré-classificada, é criada uma cópia da imagem que consiste na face isolada do restante da imagem, essa imagem por sua vez é redimensionada para se adequar ao padrão de entrada da rede CNN.

A imagem aplicada então à entrada da rede CNN que possui a função de classificar determinado usuário, produz um valor de predição que se for de acordo com a classificação positiva para tal usuário prossegue a verificação.

A ultima etapa necessária consiste na verificação da distancia euclidiana entre a face candidata e a face padrão do usuário, que, se for menor que determinado limiar, finalmente o acesso é concedido e a saída que ativa o relé é permanecerá ativa pelo tempo de 5 segundos.

2.4. *Senha alfanumérica de acesso*.

Após iniciado o programa, e se a tecla “s” for pressionada pelo tempo médio de 2 segundos, o fluxo natural do programa é direcionado para a cadeia de instruções condicionais que aguardam as teclas correspondentes a senha alfanumérica serem digitadas. Quando as teclas forem digitadas corretamente o acesso é concedido de maneira semelhante a abordada na subseção anterior. Uma vez em que o programa entrou em modo de senha, a digitação da senha pode ser interrompida e fluxo normal do programa pode ser retomado se a tecla “e” for pressionada por 2 segundos.