

TransferLearning e FineTuning di una SqueezeNet sul dataset della hands digit recognition. I parametri sono precedentemente stati trainati su 'ImageNet'.

<https://www.kaggle.com/kmader/food-squeezenet>

```
##cd
```

```
#shutil.rmtree('/content/hands_dataset', ignore_errors=True) #non usarlo
```

```
!pip install keras_applications
```

```
Collecting keras_applications
```

```
  Downloading https://files.pythonhosted.org/packages/71/e3/19762fd6c62877ae9102edf6342
```

```
    |████████████████████████████████████████| 51kB 6.8MB/s
```

```
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from ker
```

```
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (
```

```
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from h5py
```

```
Installing collected packages: keras-applications
```

```
Successfully installed keras-applications-1.0.8
```

```
from keras_applications.imagenet_utils import _obtain_input_shape
from keras import backend as K
from keras.layers import Input, Convolution2D, MaxPooling2D, Activation, concatenate, Dropout
from keras.models import Model
from keras.engine.topology import get_source_inputs
from keras.utils import get_file
from keras.utils import layer_utils
import keras
from keras.callbacks import ModelCheckpoint
```

```
import numpy as np
```

```
import math
```

```
import tensorflow as tf
```

```
#import keras.backend as K
```

```
from keras.models import Model, Sequential
```

```
from keras.layers import Input, Dense, Conv2D, Conv3D, DepthwiseConv2D, SeparableConv2D, Conv
```

```
from keras.layers import Flatten, MaxPool2D, AvgPool2D, GlobalAvgPool2D, UpSampling2D, BatchN
```

```
from keras.layers import Concatenate, Add, Dropout, ReLU, Lambda, Activation, LeakyReLU, PReLU
```

```
from math import ceil
```

```
from IPython.display import SVG
```

```
from keras.utils.vis_utils import model_to_dot
```

```
from keras.callbacks import LearningRateScheduler
```

```
from tensorflow.keras.optimizers import Adam
```

```

from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications.vgg16 import preprocess_input, decode_predictions
from sklearn.metrics import confusion_matrix
from tensorflow.keras.models import Model
from sklearn.metrics import confusion_matrix
import itertools
import itertools
import os
import shutil
import random
import glob
import matplotlib.pyplot as plt
import warnings

sq1x1 = "squeeze1x1"
exp1x1 = "expand1x1"
exp3x3 = "expand3x3"
relu = "relu_"

WEIGHTS_PATH = "https://github.com/rcmalli/keras-squeezenet/releases/download/v1.0/squeezenet"

# Modular function for Fire Node

def fire_module(x, fire_id, squeeze=16, expand=64):
    s_id = 'fire' + str(fire_id) + '/'

    if K.image_data_format() == 'channels_first':
        channel_axis = 1
    else:
        channel_axis = 3

    x = Convolution2D(squeeze, (1, 1), padding='valid', name=s_id + sq1x1)(x)
    x = Activation('relu', name=s_id + relu + sq1x1)(x)

    left = Convolution2D(expand, (1, 1), padding='valid', name=s_id + exp1x1)(x)
    left = Activation('relu', name=s_id + relu + exp1x1)(left)

    right = Convolution2D(expand, (3, 3), padding='same', name=s_id + exp3x3)(x)
    right = Activation('relu', name=s_id + relu + exp3x3)(right)

    x = concatenate([left, right], axis=channel_axis, name=s_id + 'concat')
    return x

# Original SqueezeNet from paper.

```

```

def SqueezeNet(input_tensor=None, input_shape=None,
               weights='imagenet',
               classes=1000,
               use_bn_on_input = False, # to avoid preprocessing
               first_stride = 2
               ):
    if weights not in {'imagenet', None}:
        raise ValueError('The `weights` argument should be either '
                          '`None` (random initialization) or `imagenet` '
                          '(pre-training on ImageNet).')

    if weights == 'imagenet' and classes != 1000:
        raise ValueError('If using `weights` as imagenet with `include_top` '
                          'as true, `classes` should be 1000')

    input_shape = _obtain_input_shape(input_shape,
                                       default_size=227,
                                       min_size=48,
                                       data_format=K.image_data_format(),
                                       require_flatten = False)

    if input_tensor is None:
        raw_img_input = Input(shape=input_shape)
    else:
        if not K.is_keras_tensor(input_tensor):
            img_input = Input(tensor=input_tensor, shape=input_shape)
        else:
            img_input = input_tensor
    if use_bn_on_input:
        img_input = BatchNormalization()(raw_img_input)
    else:
        img_input = raw_img_input

    x = Convolution2D(64, (3, 3), strides=(first_stride, first_stride), padding='valid', name='conv1')(img_input)
    x = Activation('relu', name='relu_conv1')(x)
    x = MaxPooling2D(pool_size=(3, 3), strides=(2, 2), name='pool1')(x)

    x = fire_module(x, fire_id=2, squeeze=16, expand=64)
    x = fire_module(x, fire_id=3, squeeze=16, expand=64)
    x = MaxPooling2D(pool_size=(3, 3), strides=(2, 2), name='pool3')(x)

    x = fire_module(x, fire_id=4, squeeze=32, expand=128)
    x = fire_module(x, fire_id=5, squeeze=32, expand=128)
    x = MaxPooling2D(pool_size=(3, 3), strides=(2, 2), name='pool5')(x)

    x = fire_module(x, fire_id=6, squeeze=48, expand=192)
    x = fire_module(x, fire_id=7, squeeze=48, expand=192)
    x = fire_module(x, fire_id=8, squeeze=64, expand=256)
    x = fire_module(x, fire_id=9, squeeze=64, expand=256)

```

```

x = Dropout(0.5, name='drop9')(x)

x = Convolution2D(classes, (1, 1), padding='valid', name='conv10')(x)
x = Activation('relu', name='relu_conv10')(x)
x = GlobalAveragePooling2D()(x)
out = Activation('softmax', name='loss')(x)

# Ensure that the model takes into account
# any potential predecessors of `input_tensor`.
if input_tensor is not None:
    inputs = get_source_inputs(input_tensor)
else:
    inputs = raw_img_input

model = Model(inputs, out, name='squeezenet')

# load weights
if weights == 'imagenet':

    weights_path = get_file('squeezenet_weights_tf_dim_ordering_tf_kernels.h5',
                             WEIGHTS_PATH,
                             cache_subdir='models')
    model.load_weights(weights_path)
    if K.backend() == 'theano':
        layer_utils.convert_all_kernels_in_model(model)

    if K.image_data_format() == 'channels_first':

        if K.backend() == 'tensorflow':
            warnings.warn('You are using the TensorFlow backend, yet you '
                          'are using the Theano '
                          'image data format convention '
                          '(`image_data_format="channels_first"`). '
                          'For best performance, set '
                          '`image_data_format="channels_last"` in '
                          'your Keras config '
                          'at ~/.keras/keras.json.')

return model

```

▼ TEST 1: training da inizializzazione casuale e senza Augmentation

(classic_model)

```

model = SqueezeNet(input_shape = (224,224,3), weights = None, classes = 10,
                   use_bn_on_input = True)
# initiate RMSprop optimizer
opt = keras.optimizers.RMSprop(lr=0.0001, decay=1e-7)

# Let's train the model using RMSprop
model.compile(loss='categorical_crossentropy'

```

```
model.compile(loss=categorical_crossentropy,
              optimizer=opt,
              metrics=['accuracy'])
```

```
model.summary()
```

Model: "squeezenet"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	
batch_normalization (BatchNormaliza	(None, 224, 224, 3)	12	input_1[0][0]
conv1 (Conv2D)	(None, 111, 111, 64)	1792	batch_normalization
relu_conv1 (Activation)	(None, 111, 111, 64)	0	conv1[0][0]
pool1 (MaxPooling2D)	(None, 55, 55, 64)	0	relu_conv1[0][0]
fire2/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	1040	pool1[0][0]
fire2/relu_squeeze1x1 (Activati	(None, 55, 55, 16)	0	fire2/squeeze1x1[0]
fire2/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire2/relu_squeeze1
fire2/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire2/relu_squeeze1
fire2/relu_expand1x1 (Activatio	(None, 55, 55, 64)	0	fire2/expand1x1[0][
fire2/relu_expand3x3 (Activatio	(None, 55, 55, 64)	0	fire2/expand3x3[0][
fire2/concat (Concatenate)	(None, 55, 55, 128)	0	fire2/relu_expand1x fire2/relu_expand3x
fire3/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	2064	fire2/concat[0][0]
fire3/relu_squeeze1x1 (Activati	(None, 55, 55, 16)	0	fire3/squeeze1x1[0]
fire3/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire3/relu_squeeze1
fire3/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire3/relu_squeeze1
fire3/relu_expand1x1 (Activatio	(None, 55, 55, 64)	0	fire3/expand1x1[0][
fire3/relu_expand3x3 (Activatio	(None, 55, 55, 64)	0	fire3/expand3x3[0][
fire3/concat (Concatenate)	(None, 55, 55, 128)	0	fire3/relu_expand1x fire3/relu_expand3x
pool3 (MaxPooling2D)	(None, 27, 27, 128)	0	fire3/concat[0][0]
fire4/squeeze1x1 (Conv2D)	(None, 27, 27, 32)	4128	pool3[0][0]
fire4/relu_squeeze1x1 (Activati	(None, 27, 27, 32)	0	fire4/squeeze1x1[0]
fire4/expand1x1 (Conv2D)	(None, 27, 27, 128)	4224	fire4/relu_squeeze1

fire4/expand3x3 (Conv2D)	(None, 27, 27, 128)	36992	fire4/relu_squeeze1
fire4/relu_expand1x1 (Activatio	(None, 27, 27, 128)	0	fire4/expand1x1[0][
fire4/relu_expand3x3 (Activatio	(None, 27, 27, 128)	0	fire4/expand3x3[0][

```
%cd
```

```
%cd ../content
```

```
!pwd
```

```
/root
```

```
/content
```

```
/content
```

```
!pwd
```

```
!git clone https://github.com/tesiiscomingson/hands_dataset.git
```

```
/content
```

```
Cloning into 'hands_dataset'...
```

```
remote: Enumerating objects: 2696, done.
```

```
remote: Total 2696 (delta 0), reused 0 (delta 0), pack-reused 2696
```

```
Receiving objects: 100% (2696/2696), 22.06 MiB | 41.15 MiB/s, done.
```

```
Resolving deltas: 100% (863/863), done.
```

```
%cd hands_dataset
```

```
!ls
```

```
/content/hands_dataset
```

```
Dataset Examples LICENSE README.md
```

```
!rm -rf Examples LICENSE README.md
```

```
!ls
```

```
Dataset
```

```
dir_path = os.path.dirname(os.path.realpath('FT_mobilenet.ipynb'))
```

```
print(dir_path)
```

```
/content/hands_dataset
```

```
!pwd
```

```
/content/hands_dataset
```

```
%cd /content/hands_dataset/Dataset
%mkdir train
%mkdir test
%mkdir valid
%mv 0/ 1/ 2 / 3/ 4/ 5/ 6/ 7/ 8/ 9/ train/

/content/hands_dataset/Dataset
mv: cannot move '/' to 'train': Device or resource busy
```

```
#%cd -
!pwd
%cd valid
%mkdir 0/ 1/ 2 / 3/ 4/ 5/ 6/ 7/ 8/ 9/
%cd ../test
%mkdir 0/ 1/ 2 / 3/ 4/ 5/ 6/ 7/ 8/ 9/

/content/hands_dataset/Dataset
/content/hands_dataset/Dataset/valid
mkdir: cannot create directory '/': File exists
/content/hands_dataset/Dataset/test
mkdir: cannot create directory '/': File exists
```

```
!pwd
```

```
/content/hands_dataset/Dataset/test
```

```
%%bash
cd ../train
for ((i=0; i<=9; i++)); do
  a=$(find $i/ -type f | shuf -n 30)
  mv $a ../valid/$i/
  b=$(find $i/ -type f | shuf -n 5)
  mv $b ../test/$i/
done
```

```
%cd ../..
!pwd
```

```
/content/hands_dataset
/content/hands_dataset
```

```
train_path = 'Dataset/train'
valid_path = 'Dataset/valid'
test_path = 'Dataset/test'
```

```
b_size=16
```

```

train_batches = ImageDataGenerator(preprocessing_function=keras.applications.mobilenet.preproc
train_path, target_size=(224, 224), batch_size=b_size)
valid_batches = ImageDataGenerator(preprocessing_function=keras.applications.mobilenet.preproc
valid_path, target_size=(224, 224), batch_size=b_size)
test_batches = ImageDataGenerator(preprocessing_function=keras.applications.mobilenet.preproc
test_path, target_size=(224, 224), batch_size=b_size, shuffle=False)

```

```

Found 2172 images belonging to 10 classes.
Found 300 images belonging to 10 classes.
Found 50 images belonging to 10 classes.

```

```
metrics = ['accuracy']
```

```

train_steps=ceil(2172/b_size)
valid_steps=ceil(300/b_size)

```

```
history = model.fit_generator(train_batches, steps_per_epoch=train_steps, validation_data=valid_batches, validation_steps=valid_steps)
```

```

/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:184:
warnings.warn("`Model.fit_generator` is deprecated and will be removed in a future version.")
Epoch 1/50
136/136 - 42s - loss: 2.3028 - accuracy: 0.0889 - val_loss: 2.3026 - val_accuracy: 0.0889
Epoch 2/50
136/136 - 6s - loss: 2.3027 - accuracy: 0.0976 - val_loss: 2.3026 - val_accuracy: 0.0976
Epoch 3/50
136/136 - 6s - loss: 2.3028 - accuracy: 0.1013 - val_loss: 2.3023 - val_accuracy: 0.1013
Epoch 4/50
136/136 - 6s - loss: 2.2972 - accuracy: 0.0999 - val_loss: 2.3055 - val_accuracy: 0.0999
Epoch 5/50
136/136 - 6s - loss: 2.2888 - accuracy: 0.0994 - val_loss: 2.2855 - val_accuracy: 0.0994
Epoch 6/50
136/136 - 6s - loss: 2.2691 - accuracy: 0.1105 - val_loss: 2.2488 - val_accuracy: 0.1105
Epoch 7/50
136/136 - 6s - loss: 2.2143 - accuracy: 0.1344 - val_loss: 2.1107 - val_accuracy: 0.1344
Epoch 8/50
136/136 - 6s - loss: 2.1031 - accuracy: 0.1750 - val_loss: 2.3106 - val_accuracy: 0.1750
Epoch 9/50
136/136 - 6s - loss: 2.0402 - accuracy: 0.2058 - val_loss: 1.9348 - val_accuracy: 0.2058
Epoch 10/50
136/136 - 6s - loss: 1.9545 - accuracy: 0.2362 - val_loss: 1.8570 - val_accuracy: 0.2362
Epoch 11/50
136/136 - 6s - loss: 1.8447 - accuracy: 0.2887 - val_loss: 1.7200 - val_accuracy: 0.2887
Epoch 12/50
136/136 - 6s - loss: 1.7354 - accuracy: 0.3393 - val_loss: 1.7596 - val_accuracy: 0.3393
Epoch 13/50
136/136 - 6s - loss: 1.6256 - accuracy: 0.3973 - val_loss: 1.5847 - val_accuracy: 0.3973
Epoch 14/50
136/136 - 6s - loss: 1.5102 - accuracy: 0.4434 - val_loss: 1.3800 - val_accuracy: 0.4434
Epoch 15/50
136/136 - 6s - loss: 1.3969 - accuracy: 0.4899 - val_loss: 1.4367 - val_accuracy: 0.4899
Epoch 16/50
136/136 - 6s - loss: 1.2698 - accuracy: 0.5387 - val_loss: 1.1780 - val_accuracy: 0.5387

```



```

Epoch 17/50
136/136 - 6s - loss: 1.2117 - accuracy: 0.5552 - val_loss: 1.0174 - val_accuracy: 0.
Epoch 18/50
136/136 - 6s - loss: 1.1215 - accuracy: 0.5985 - val_loss: 0.9010 - val_accuracy: 0.
Epoch 19/50
136/136 - 6s - loss: 1.0233 - accuracy: 0.6289 - val_loss: 0.8580 - val_accuracy: 0.
Epoch 20/50
136/136 - 6s - loss: 0.9652 - accuracy: 0.6602 - val_loss: 0.9263 - val_accuracy: 0.
Epoch 21/50
136/136 - 6s - loss: 0.8602 - accuracy: 0.6888 - val_loss: 0.7815 - val_accuracy: 0.
Epoch 22/50
136/136 - 6s - loss: 0.7946 - accuracy: 0.7086 - val_loss: 0.6899 - val_accuracy: 0.
Epoch 23/50
136/136 - 6s - loss: 0.7321 - accuracy: 0.7523 - val_loss: 0.6186 - val_accuracy: 0.
Epoch 24/50
136/136 - 6s - loss: 0.6393 - accuracy: 0.7772 - val_loss: 0.6152 - val_accuracy: 0.
Epoch 25/50
136/136 - 6s - loss: 0.6085 - accuracy: 0.7951 - val_loss: 0.4995 - val_accuracy: 0.
Epoch 26/50
136/136 - 6s - loss: 0.5375 - accuracy: 0.8255 - val_loss: 0.4800 - val_accuracy: 0.
Epoch 27/50
136/136 - 6s - loss: 0.4877 - accuracy: 0.8430 - val_loss: 0.7004 - val_accuracy: 0.
Epoch 28/50
136/136 - 6s - loss: 0.4502 - accuracy: 0.8620 - val_loss: 0.3401 - val_accuracy: 0.

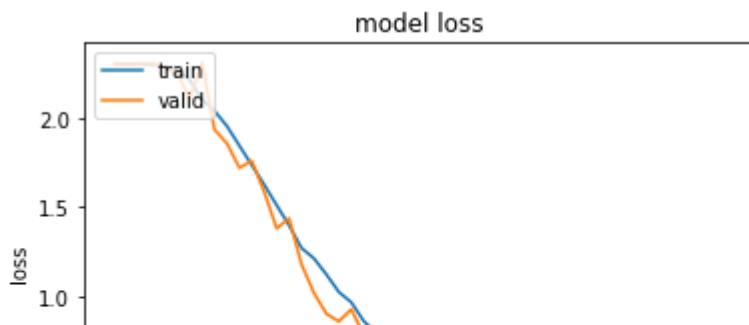
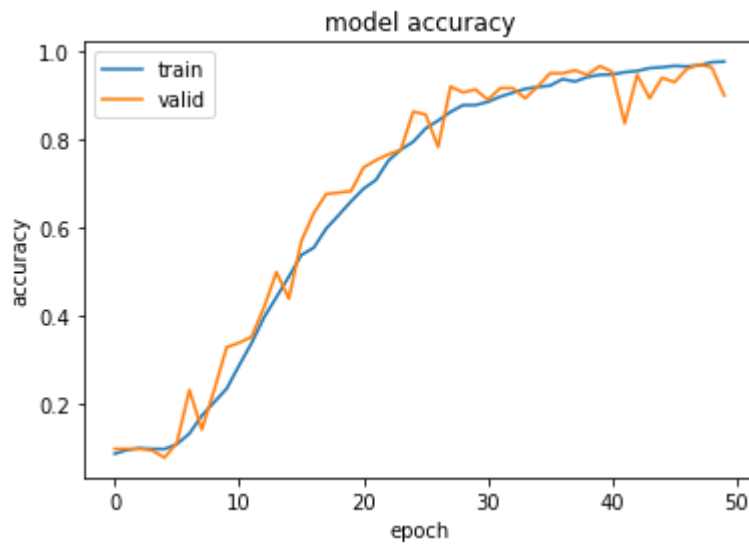
```

```

def sum_up(history):
    # summarize history for accuracy
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('model accuracy')
    plt.ylabel('accuracy')
    plt.xlabel('epoch')
    plt.legend(['train', 'valid'], loc='upper left')
    plt.show()
    # summarize history for loss
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train', 'valid'], loc='upper left')
    plt.show()

```

```
sum_up(history)
```



```
model.save('classic_squeeze.h5')
```

```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
```

```

plt.text(j, i, cm[i, j],
        horizontalalignment="center",
        color="white" if cm[i, j] > thresh else "black")

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')

```

```
test_labels = test_batches.classes
```

```
predictions = model.predict_generator(test_batches, steps=ceil(50/b_size), verbose=0)
```

```

/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:1905:
warnings.warn("`Model.predict_generator` is deprecated and '

```

```
cm = confusion_matrix(test_labels, predictions.argmax(axis=1))
```

```
cm_plot_labels = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
```

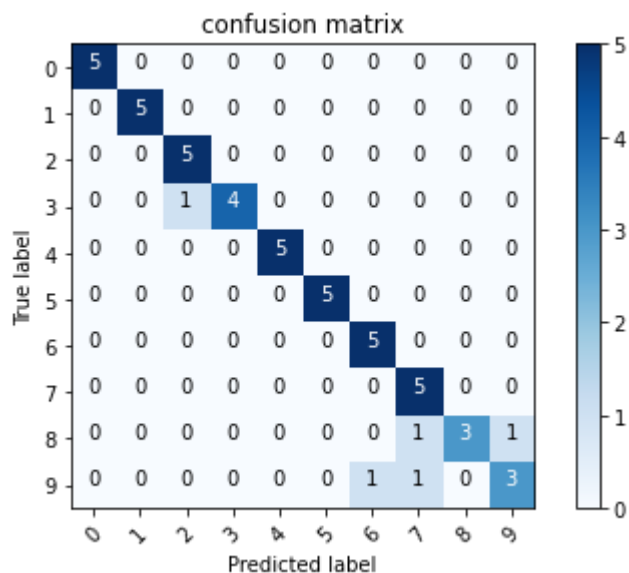
```
plot_confusion_matrix(cm, cm_plot_labels, title='confusion matrix')
```

Confusion matrix, without normalization

```

[[5 0 0 0 0 0 0 0 0 0]
 [0 5 0 0 0 0 0 0 0 0]
 [0 0 5 0 0 0 0 0 0 0]
 [0 0 1 4 0 0 0 0 0 0]
 [0 0 0 0 5 0 0 0 0 0]
 [0 0 0 0 0 5 0 0 0 0]
 [0 0 0 0 0 0 5 0 0 0]
 [0 0 0 0 0 0 0 5 0 0]
 [0 0 0 0 0 0 0 0 5 0]
 [0 0 0 0 0 0 0 1 3 1]
 [0 0 0 0 0 0 1 1 0 3]]

```



Nonostante l'apparente successo in fase di test con webcam la rete oscilla sempre tra zero e uno. Potrebbe significare che è in grado di riconoscere solo immagini simili al dataset su cui è avvenuto il training, proviamo con un nuovo modello con Augmentation

```
ft_model = SqueezeNet(input_shape = (224,224,3), weights = 'imagenet', classes = 1000,
                      use_bn_on_input = False)
```

Downloading data from <https://github.com/rcmalli/keras-squeezenet/releases/download/v1.5062656/5059384> [=====] - 0s 0us/step

```
ft_model.summary()
```

Model: "squeezenet"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	
conv1 (Conv2D)	(None, 111, 111, 64)	1792	input_2[0][0]
relu_conv1 (Activation)	(None, 111, 111, 64)	0	conv1[0][0]
pool1 (MaxPooling2D)	(None, 55, 55, 64)	0	relu_conv1[0][0]
fire2/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	1040	pool1[0][0]
fire2/relu_squeeze1x1 (Activation)	(None, 55, 55, 16)	0	fire2/squeeze1x1[0]
fire2/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire2/relu_squeeze1
fire2/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire2/relu_squeeze1
fire2/relu_expand1x1 (Activation)	(None, 55, 55, 64)	0	fire2/expand1x1[0]
fire2/relu_expand3x3 (Activation)	(None, 55, 55, 64)	0	fire2/expand3x3[0]
fire2/concat (Concatenate)	(None, 55, 55, 128)	0	fire2/relu_expand1x1 fire2/relu_expand3x3
fire3/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	2064	fire2/concat[0][0]
fire3/relu_squeeze1x1 (Activation)	(None, 55, 55, 16)	0	fire3/squeeze1x1[0]
fire3/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire3/relu_squeeze1
fire3/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire3/relu_squeeze1
fire3/relu_expand1x1 (Activation)	(None, 55, 55, 64)	0	fire3/expand1x1[0]
fire3/relu_expand3x3 (Activation)	(None, 55, 55, 64)	0	fire3/expand3x3[0]
fire3/concat (Concatenate)	(None, 55, 55, 128)	0	fire3/relu_expand1x1 fire3/relu_expand3x3

pool3 (MaxPooling2D)	(None, 27, 27, 128)	0	fire3/concat[0][0]
fire4/squeeze1x1 (Conv2D)	(None, 27, 27, 32)	4128	pool3[0][0]
fire4/relu_squeeze1x1 (Activation)	(None, 27, 27, 32)	0	fire4/squeeze1x1[0]
fire4/expand1x1 (Conv2D)	(None, 27, 27, 128)	4224	fire4/relu_squeeze1
fire4/expand3x3 (Conv2D)	(None, 27, 27, 128)	36992	fire4/relu_squeeze1
fire4/relu_expand1x1 (Activation)	(None, 27, 27, 128)	0	fire4/expand1x1[0][
fire4/relu_expand3x3 (Activation)	(None, 27, 27, 128)	0	fire4/expand3x3[0][
fire4/concat (Concatenate)	(None, 27, 27, 256)	0	fire4/relu_expand1x

```
x = Sequential()
x = ft_model.layers[-6].output
# we take out only the last 4 layers
```

```
x = (Convolution2D(10, (1, 1), padding='valid', name='conv10'))(x)
x = (Activation('relu', name='relu_conv10'))(x)
x = (GlobalAveragePooling2D())(x)
pred = (Activation('softmax', name='loss'))(x)
new_model = Model(inputs=ft_model.input, outputs=pred)
new_model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	
conv1 (Conv2D)	(None, 111, 111, 64)	1792	input_2[0][0]
relu_conv1 (Activation)	(None, 111, 111, 64)	0	conv1[0][0]
pool1 (MaxPooling2D)	(None, 55, 55, 64)	0	relu_conv1[0][0]
fire2/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	1040	pool1[0][0]
fire2/relu_squeeze1x1 (Activation)	(None, 55, 55, 16)	0	fire2/squeeze1x1[0]
fire2/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire2/relu_squeeze1
fire2/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire2/relu_squeeze1
fire2/relu_expand1x1 (Activation)	(None, 55, 55, 64)	0	fire2/expand1x1[0][
fire2/relu_expand3x3 (Activation)	(None, 55, 55, 64)	0	fire2/expand3x3[0][
fire2/concat (Concatenate)	(None, 55, 55, 128)	0	fire2/relu_expand1x

			fire2/relu_expand3x
fire3/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	2064	fire2/concat[0][0]
fire3/relu_squeeze1x1 (Activation)	(None, 55, 55, 16)	0	fire3/squeeze1x1[0]
fire3/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire3/relu_squeeze1
fire3/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire3/relu_squeeze1
fire3/relu_expand1x1 (Activation)	(None, 55, 55, 64)	0	fire3/expand1x1[0]
fire3/relu_expand3x3 (Activation)	(None, 55, 55, 64)	0	fire3/expand3x3[0]
fire3/concat (Concatenate)	(None, 55, 55, 128)	0	fire3/relu_expand1x fire3/relu_expand3x
pool3 (MaxPooling2D)	(None, 27, 27, 128)	0	fire3/concat[0][0]
fire4/squeeze1x1 (Conv2D)	(None, 27, 27, 32)	4128	pool3[0][0]
fire4/relu_squeeze1x1 (Activation)	(None, 27, 27, 32)	0	fire4/squeeze1x1[0]
fire4/expand1x1 (Conv2D)	(None, 27, 27, 128)	4224	fire4/relu_squeeze1
fire4/expand3x3 (Conv2D)	(None, 27, 27, 128)	36992	fire4/relu_squeeze1
fire4/relu_expand1x1 (Activation)	(None, 27, 27, 128)	0	fire4/expand1x1[0]
fire4/relu_expand3x3 (Activation)	(None, 27, 27, 128)	0	fire4/expand3x3[0]
fire4/concat (Concatenate)	(None, 27, 27, 256)	0	fire4/relu_expand1x fire4/relu_expand3x

```
for layer in new_model.layers[:-25]:
    layer.trainable = False
```

```
new_model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	
conv1 (Conv2D)	(None, 111, 111, 64)	1792	input_2[0][0]
relu_conv1 (Activation)	(None, 111, 111, 64)	0	conv1[0][0]
pool1 (MaxPooling2D)	(None, 55, 55, 64)	0	relu_conv1[0][0]
fire2/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	1040	pool1[0][0]
fire2/relu_squeeze1x1 (Activation)	(None, 55, 55, 16)	0	fire2/squeeze1x1[0]

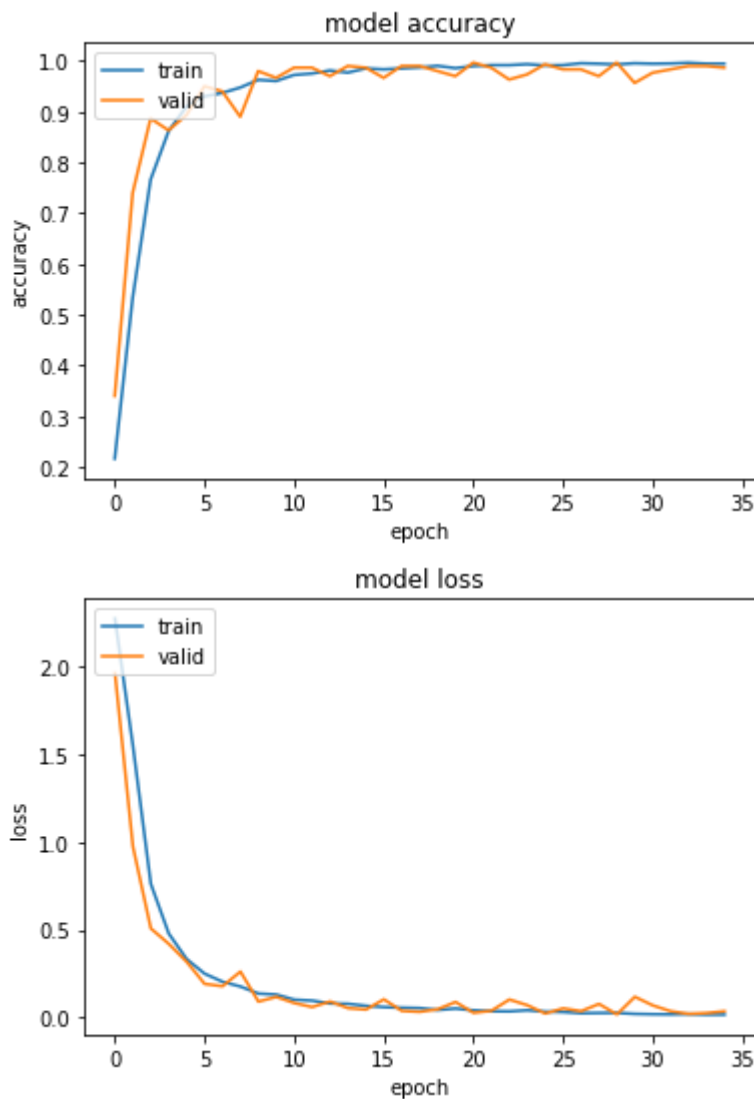
fire2/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire2/relu_squeeze1
fire2/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire2/relu_squeeze1
fire2/relu_expand1x1 (Activatio	(None, 55, 55, 64)	0	fire2/expand1x1[0][
fire2/relu_expand3x3 (Activatio	(None, 55, 55, 64)	0	fire2/expand3x3[0][
fire2/concat (Concatenate)	(None, 55, 55, 128)	0	fire2/relu_expand1x fire2/relu_expand3x
fire3/squeeze1x1 (Conv2D)	(None, 55, 55, 16)	2064	fire2/concat[0][0]
fire3/relu_squeeze1x1 (Activati	(None, 55, 55, 16)	0	fire3/squeeze1x1[0]
fire3/expand1x1 (Conv2D)	(None, 55, 55, 64)	1088	fire3/relu_squeeze1
fire3/expand3x3 (Conv2D)	(None, 55, 55, 64)	9280	fire3/relu_squeeze1
fire3/relu_expand1x1 (Activatio	(None, 55, 55, 64)	0	fire3/expand1x1[0][
fire3/relu_expand3x3 (Activatio	(None, 55, 55, 64)	0	fire3/expand3x3[0][
fire3/concat (Concatenate)	(None, 55, 55, 128)	0	fire3/relu_expand1x fire3/relu_expand3x
pool3 (MaxPooling2D)	(None, 27, 27, 128)	0	fire3/concat[0][0]
fire4/squeeze1x1 (Conv2D)	(None, 27, 27, 32)	4128	pool3[0][0]
fire4/relu_squeeze1x1 (Activati	(None, 27, 27, 32)	0	fire4/squeeze1x1[0]
fire4/expand1x1 (Conv2D)	(None, 27, 27, 128)	4224	fire4/relu_squeeze1
fire4/expand3x3 (Conv2D)	(None, 27, 27, 128)	36992	fire4/relu_squeeze1
fire4/relu_expand1x1 (Activatio	(None, 27, 27, 128)	0	fire4/expand1x1[0][
fire4/relu_expand3x3 (Activatio	(None, 27, 27, 128)	0	fire4/expand3x3[0][
fire4/concat (Concatenate)	(None, 27, 27, 256)	0	fire4/relu_expand1x

```
# initiate RMSprop optimizer
opt = keras.optimizers.RMSprop(lr=0.0001, decay=1e-7)
mc = ModelCheckpoint('best_model.h5', monitor='val_accuracy', mode='max', save_best_only=True)
# Let's train the model using RMSprop
new_model.compile(loss='categorical_crossentropy',
                  optimizer=opt,
                  metrics=['accuracy'])
history = new_model.fit_generator(train_batches, steps_per_epoch=train_steps, validation_data=
```

```
Epoch 1/35
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:18:
  warnings.warn("`Model.fit_generator` is deprecated and '
136/136 - 7s - loss: 2.2753 - accuracy: 0.2164 - val_loss: 1.9642 - val_accuracy: 0.
Epoch 2/35
136/136 - 6s - loss: 1.5509 - accuracy: 0.5331 - val_loss: 0.9755 - val_accuracy: 0.
Epoch 3/35
136/136 - 6s - loss: 0.7622 - accuracy: 0.7661 - val_loss: 0.5080 - val_accuracy: 0.
Epoch 4/35
136/136 - 6s - loss: 0.4787 - accuracy: 0.8614 - val_loss: 0.4201 - val_accuracy: 0.
Epoch 5/35
136/136 - 6s - loss: 0.3328 - accuracy: 0.9102 - val_loss: 0.3176 - val_accuracy: 0.
Epoch 6/35
136/136 - 6s - loss: 0.2500 - accuracy: 0.9300 - val_loss: 0.1916 - val_accuracy: 0.
Epoch 7/35
136/136 - 6s - loss: 0.2037 - accuracy: 0.9369 - val_loss: 0.1784 - val_accuracy: 0.
Epoch 8/35
136/136 - 6s - loss: 0.1765 - accuracy: 0.9475 - val_loss: 0.2618 - val_accuracy: 0.
Epoch 9/35
136/136 - 6s - loss: 0.1365 - accuracy: 0.9627 - val_loss: 0.0907 - val_accuracy: 0.
Epoch 10/35
136/136 - 6s - loss: 0.1306 - accuracy: 0.9604 - val_loss: 0.1173 - val_accuracy: 0.
Epoch 11/35
136/136 - 6s - loss: 0.1014 - accuracy: 0.9724 - val_loss: 0.0825 - val_accuracy: 0.
Epoch 12/35
136/136 - 6s - loss: 0.0961 - accuracy: 0.9751 - val_loss: 0.0596 - val_accuracy: 0.
Epoch 13/35
136/136 - 6s - loss: 0.0800 - accuracy: 0.9811 - val_loss: 0.0911 - val_accuracy: 0.
Epoch 14/35
136/136 - 6s - loss: 0.0780 - accuracy: 0.9770 - val_loss: 0.0533 - val_accuracy: 0.
Epoch 15/35
136/136 - 6s - loss: 0.0664 - accuracy: 0.9853 - val_loss: 0.0454 - val_accuracy: 0.
Epoch 16/35
136/136 - 6s - loss: 0.0592 - accuracy: 0.9834 - val_loss: 0.1032 - val_accuracy: 0.
Epoch 17/35
136/136 - 6s - loss: 0.0556 - accuracy: 0.9853 - val_loss: 0.0376 - val_accuracy: 0.
Epoch 18/35
136/136 - 6s - loss: 0.0535 - accuracy: 0.9871 - val_loss: 0.0324 - val_accuracy: 0.
Epoch 19/35
136/136 - 6s - loss: 0.0427 - accuracy: 0.9903 - val_loss: 0.0469 - val_accuracy: 0.
Epoch 20/35
136/136 - 6s - loss: 0.0511 - accuracy: 0.9857 - val_loss: 0.0883 - val_accuracy: 0.
Epoch 21/35
136/136 - 6s - loss: 0.0407 - accuracy: 0.9899 - val_loss: 0.0266 - val_accuracy: 0.
Epoch 22/35
136/136 - 6s - loss: 0.0356 - accuracy: 0.9917 - val_loss: 0.0384 - val_accuracy: 0.
Epoch 23/35
136/136 - 6s - loss: 0.0349 - accuracy: 0.9917 - val_loss: 0.1023 - val_accuracy: 0.
Epoch 24/35
136/136 - 6s - loss: 0.0411 - accuracy: 0.9936 - val_loss: 0.0700 - val_accuracy: 0.
Epoch 25/35
136/136 - 6s - loss: 0.0334 - accuracy: 0.9913 - val_loss: 0.0228 - val_accuracy: 0.
Epoch 26/35
136/136 - 6s - loss: 0.0311 - accuracy: 0.9917 - val_loss: 0.0518 - val_accuracy: 0.
Epoch 27/35
136/136 - 6s - loss: 0.0246 - accuracy: 0.9954 - val_loss: 0.0360 - val_accuracy: 0.
Epoch 28/35
```



```
sum_up(history)
```



```
test_labels = test_batches.classes
predictions = new_model.predict_generator(test_batches, steps=ceil(50/b_size), verbose=0)
```

/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/training.py:1905:
warnings.warn("`Model.predict_generator` is deprecated and "

```
cm = confusion_matrix(test_labels, predictions.argmax(axis=1))
```

```
cm_plot_labels = ['0','1','2','3','4','5','6','7','8','9']
plot_confusion_matrix(cm, cm_plot_labels, title='confusion matrix')
```

Confusion matrix, without normalization

```
[[5 0 0 0 0 0 0 0 0 0]
 [0 5 0 0 0 0 0 0 0 0]
 [0 0 5 0 0 0 0 0 0 0]
 [0 0 0 5 0 0 0 0 0 0]
 [0 0 0 0 5 0 0 0 0 0]
 [0 0 0 0 0 5 0 0 0 0]
 [0 0 0 0 0 0 5 0 0 0]
 [0 0 0 0 0 0 0 5 1 0]
 [0 0 0 0 0 0 0 0 5 0]
 [0 0 0 0 0 0 0 0 0 5]]
```

