

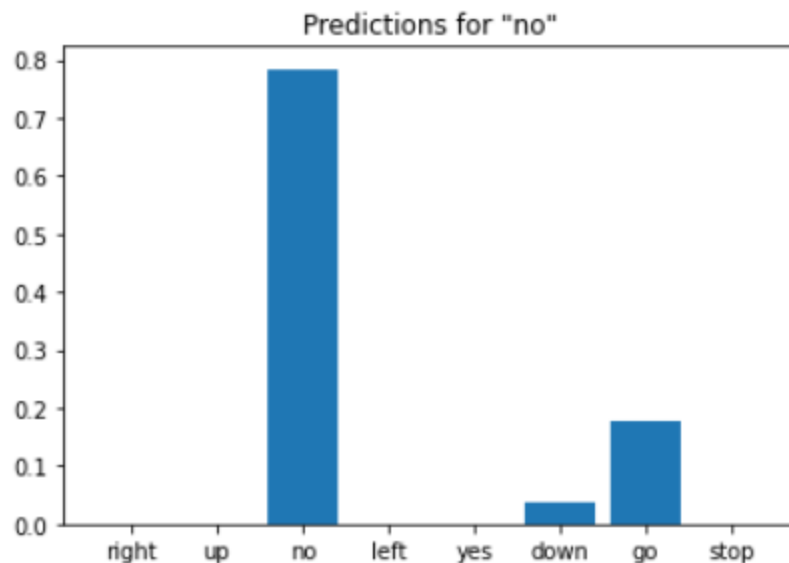
Laporan Analisis Pemrosesan Suara

1. Tulisan eksplorasi, apa kekurangan dari metode dengan mengubah konversi data menjadi image pada tutorial yang diberikan

Diberikan sebuah model yang mencoba untuk mengklasifikasikan klip audio satu detik sebagai berikut "down", "go", "left", "no", "right", "stop", "up" and "yes".

a. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan "no".
Seberapa baik performasinya?

```
sample_file = data_dir/'no/01bb6a2a_nohash_0.wav'  
  
sample_ds = preprocess_dataset([str(sample_file)])  
|  
for spectrogram, label in sample_ds.batch(1):  
    prediction = model(spectrogram)  
    plt.bar(commands, tf.nn.softmax(prediction[0]))  
    plt.title(f'Predictions for "{commands[label[0]]}"')  
    plt.show()
```



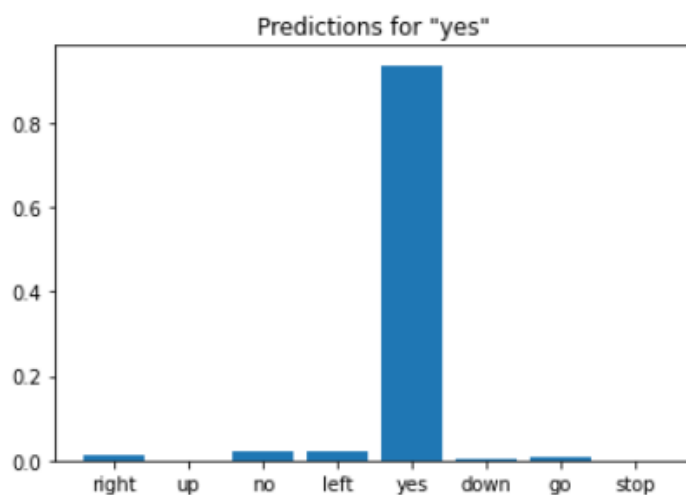
Berdasarkan hasil prediksi suara “no” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020), dan *command* no dikenali sebagai *command* go dan down dengan nilai prediksi > 0 karena terdapat missklasifikasi disebabkan pelafalannya go dan down dikenali mirip dengan no karena mempunyai huruf vocal yang sama yaitu huruf ‘o’.

b. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan "**yes**".
Seberapa baik performasinya?

```
sample_file = data_dir/'yes/00f0204f_nohash_0.wav'

sample_ds = preprocess_dataset([str(sample_file)])

for spectrogram, label in sample_ds.batch(1):
    prediction = model(spectrogram)
    plt.bar(commands, tf.nn.softmax(prediction[0]))
    plt.title(f'Predictions for "{commands[label[0]]}"')
    plt.show()
```



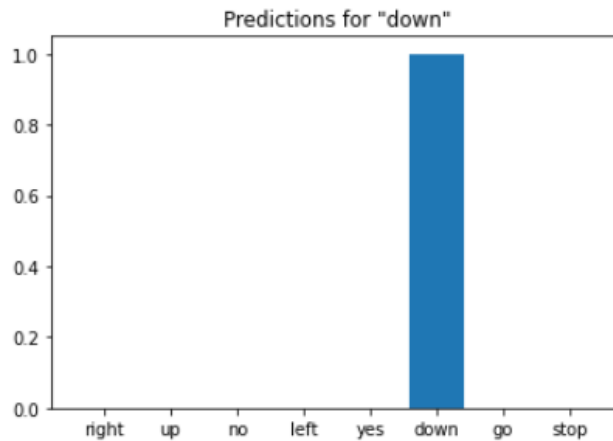
Berdasarkan hasil prediksi suara “yes” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020). Pada grafik diatas yang dilakukan untuk audio dengan suara “yes” dapat memprediksi dengan benar bahkan mencapai akurasi lebih dari 0.8%.

c. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan "**down**".
Seberapa baik performasinya?

```
sample_file = data_dir/'down/00b01445_nohash_1.wav'

sample_ds = preprocess_dataset([str(sample_file)])

for spectrogram, label in sample_ds.batch(1):
    prediction = model(spectrogram)
    plt.bar(commands, tf.nn.softmax(prediction[0]))
    plt.title(f'Predictions for "{commands[label[0]]}"')
    plt.show()
```



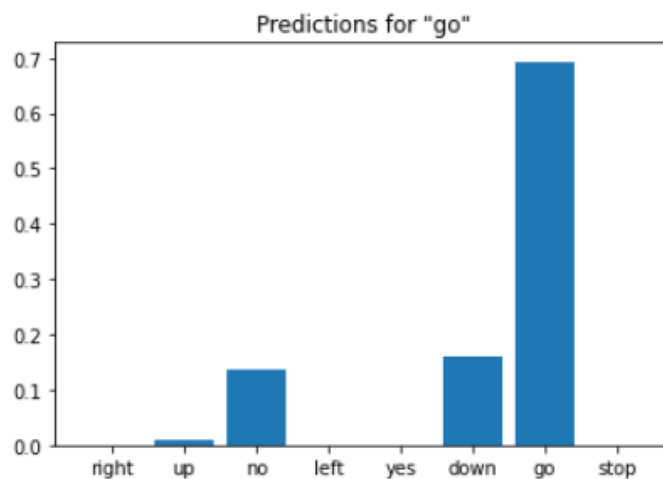
Berdasarkan hasil prediksi suara “down” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020). Dapat memprediksi dengan benar bahkan mencapai akurasi 1.0%. artinya dapat memprediksi suara “down” sangat baik

d. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan "go". Seberapa baik performasinya?

```
sample_file = data_dir/'go/004ae714_nohash_0.wav'

sample_ds = preprocess_dataset([str(sample_file)])

for spectrogram, label in sample_ds.batch(1):
    prediction = model(spectrogram)
    plt.bar(commands, tf.nn.softmax(prediction[0]))
    plt.title(f'Predictions for "{commands[label[0]]}"')
    plt.show()
```



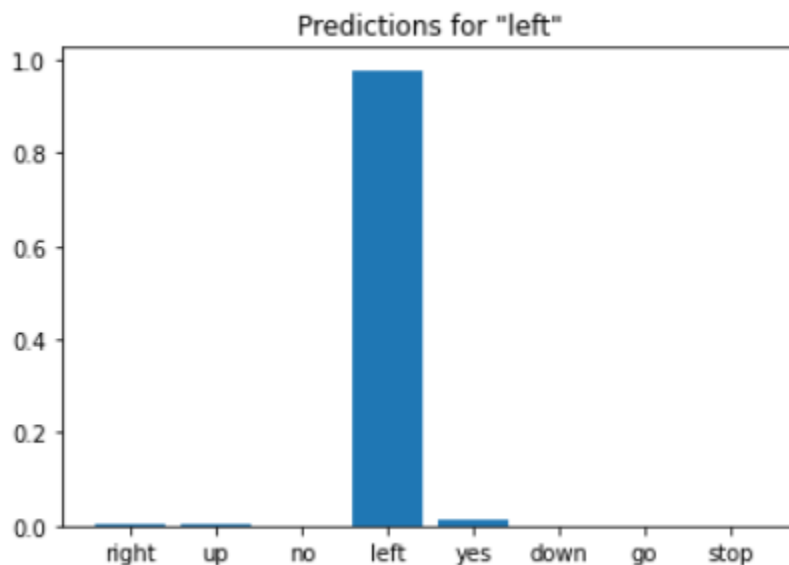
Berdasarkan hasil prediksi suara “go” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020). Dapat memprediksi dengan benar bahkan mencapai akurasi lebih dari 0.7%. artinya dapat memprediksi suara “go” sangat baik. Akan tetapi terdapat missklasifikasi disebabkan pelafalannya go dan down dikenali mirip dengan no karena mempunyai huruf vocal yang sama yaitu huruf ‘o’

e. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan “left”. Seberapa baik performasinya?

```
sample_file = data_dir/'left/012c8314_nohash_0.wav'

sample_ds = preprocess_dataset([str(sample_file)])

for spectrogram, label in sample_ds.batch(1):
    prediction = model(spectrogram)
    plt.bar(commands, tf.nn.softmax(prediction[0]))
    plt.title(f'Predictions for "{commands[label[0]]}"')
    plt.show()
```



Berdasarkan hasil prediksi suara “left” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020).

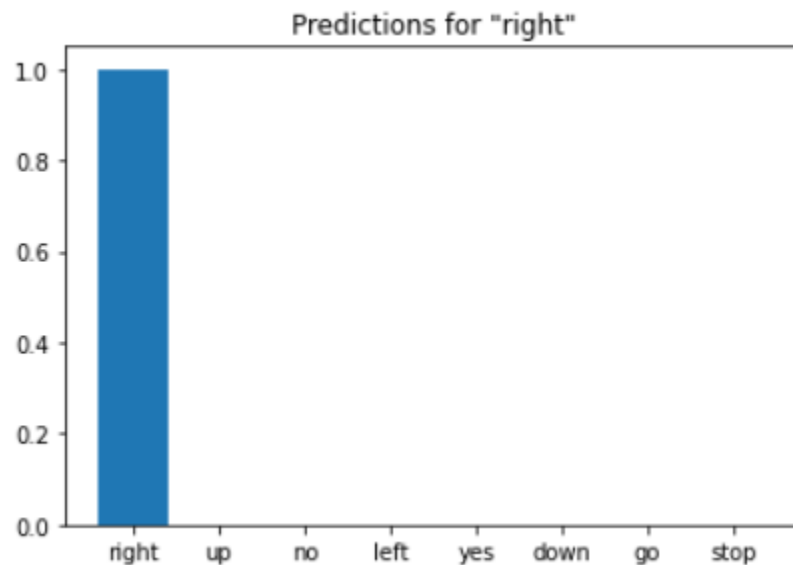
Dapat memprediksi dengan benar bahkan mencapai akurasi 1.0%. artinya dapat memprediksi suara “left” sangat baik

f. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan "**right**". Seberapa baik performasinya?

```
sample_file = data_dir/'right/0132a06d_nohash_1.wav'

sample_ds = preprocess_dataset([str(sample_file)])

for spectrogram, label in sample_ds.batch(1):
    prediction = model(spectrogram)
    plt.bar(commands, tf.nn.softmax(prediction[0]))
    plt.title(f'Predictions for "{commands[label[0]]}"')
    plt.show()
```



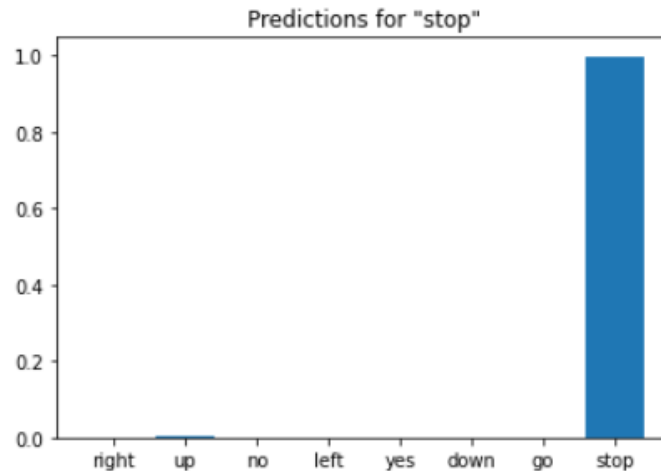
Berdasarkan hasil prediksi suara “right” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020). Dapat memprediksi dengan benar bahkan mencapai akurasi 1.0%. artinya dapat memprediksi suara “right” sangat baik

g. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan "**stop**". Seberapa baik performasinya?

```
sample_file = data_dir/'stop/012c8314_nohash_0.wav'

sample_ds = preprocess_dataset([str(sample_file)])

for spectrogram, label in sample_ds.batch(1):
    prediction = model(spectrogram)
    plt.bar(commands, tf.nn.softmax(prediction[0]))
    plt.title(f'Predictions for "{commands[label[0]]}"')
    plt.show()
```



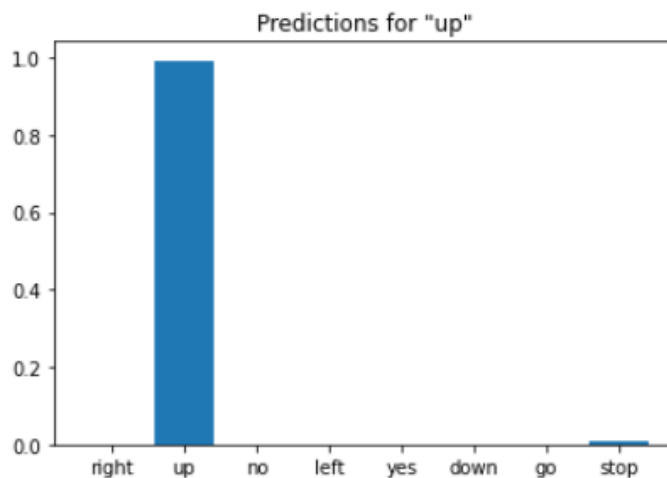
Berdasarkan hasil prediksi suara “stop” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020). Dapat memprediksi dengan benar bahkan mencapai akurasi 1.0%. artinya dapat memprediksi suara “stop” sangat baik

h. Prediksi model menggunakan file audio masukan dari seseorang yang mengatakan "**up**". Seberapa baik performasinya?

```
sample_file = data_dir/'up/0132a06d_nohash_2.wav'

sample_ds = preprocess_dataset([str(sample_file)])

for spectrogram, label in sample_ds.batch(1):
    prediction = model(spectrogram)
    plt.bar(commands, tf.nn.softmax(prediction[0]))
    plt.title(f'Predictions for "{commands[label[0]]}"')
    plt.show()
```

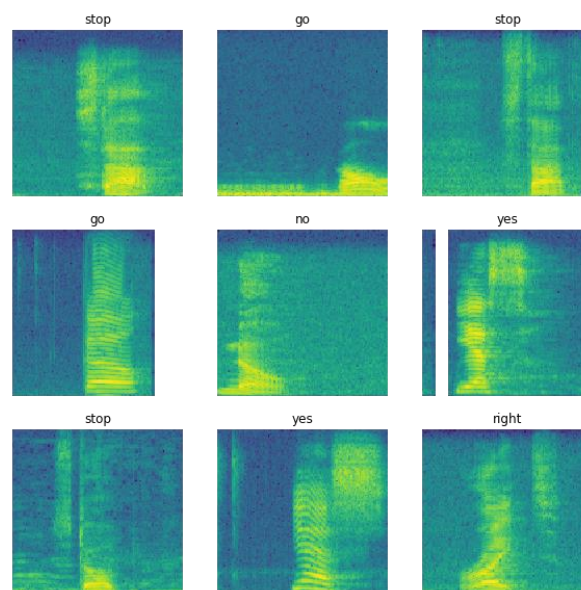


Berdasarkan hasil prediksi suara “up” konversi data menjadi image output yang dihasilkan menghasilkan nilai prediksi > 0 itu terjadi karena tensorflow memberikan cara yang lebih baik untuk memvisualisasikan data dengan pendekatan grafisnya. Ini juga memungkinkan proses debug node yang mudah dengan bantuan TensorBoard. Ini mengurangi upaya mengunjungi seluruh kode dan secara efektif menyelesaikan neural network (techvidvan, 2020). Dapat memprediksi dengan benar bahkan mencapai akurasi 1.0%. artinya dapat memprediksi suara “up” sangat baik

2. Berdasarkan hasil percobaan yang telah saya lakukan terdapat kekurangan dari metode dengan mengubah konversi data menjadi image pada tutorial yang diberikan pada suara “go”, “down” dan “no” karena salah mengklasifikasikannya pada ke-3 suara tersebut yang disebabkan pada proses mengubah bentuk gelombang menjadi spectrogram, yang menunjukkan perubahan frekuensi dari waktu ke waktu dan dapat direpresentasikan sebagai gambar 2D

Now transform the waveform dataset to have spectrogram images and their corresponding labels as integer IDs.

```
[15] def get_spectrogram_and_label_id(audio, label):  
    spectrogram = get_spectrogram(audio)  
    spectrogram = tf.expand_dims(spectrogram, -1)  
    label_id = tf.argmax(label == commands)  
    return spectrogram, label_id  
  
[16] spectrogram_ds = waveform_ds.map(  
    get_spectrogram_and_label_id, num_parallel_calls=AUTOTUNE)
```



Karena saat audio diubah menjadi spasial 2D akan diubah menjadi segmen yang diperlakukan sebagai vector angka dan vector tersebut disusun dalam urutan waktu untuk membentuk array dua dimensi. Karena perubahan urutan memori TensorFlow, waktu dalam gambar ini meningkat dari atas ke bawah, dengan frekuensi bergerak dari kiri ke kanan. Fitur spectrogram untuk fitur konsonan dapat direpresentasikan dengan lebih baik, sedangkan pada fitur vocal kurang dapat merepresentasikan sehingga terjadi permasalahan pada huruf vocal yang menyebabkan prediksi kurang tepat.