

Sistem Pengenalan Rambu Lalu Lintas

Antonius Rildo, Alfeto, Chaterine Cristianti, Adryansyach Prasetya, Caesario
Teknik Komputer, Universitas Multimedia Nusantara, Tangerang Selatan, Indonesia

Abstrak—Deep Learning adalah sebuah cabang dari bidang Machine Learning. Visi komputer, tepatnya klasifikasi objek dapat dilakukan dengan sangat baik dengan menggunakan Deep Learning. CNN adalah salah satu metode yang dapat digunakan untuk klasifikasi citra objek. Salah satu pemanfaatan klasifikasi citra adalah dengan membuat sistem yang dapat mengenali dan mengklasifikasi rambu lalu lintas yang berada di jalan.

Kata Kunci— Deep learning, Convolution Neural Network, Rambu Lalu Lintas.

I. PENDAHULUAN

Seiring berkembangnya teknologi dan memasuki zaman industri otomotif yang semakin meluas, self-driving cars menjadi salah satu solusi yang dapat dijanjikan oleh banyak perusahaan. Bahkan bukan hanya mengenai itu, dalam project yang kami lampirkan ini juga dapat membantu pengendara yang menggunakan transportasi bermotor. Besarnya hype yang tinggi dari para pecinta otomotif setelah keluarnya beberapa seri dari perusahaan Tesla membuat pasar mobil yang dapat berjalan sendiri terlihat begitu menggiurkan. Tetapi, perkembangan autonomous vehicle masih belum terlihat begitu besar. Terlepas dari usaha dan kerja keras berbagai perusahaan besar di bidang teknologi, tampaknya konsep fully autonomous vehicle tampaknya masih sangat jauh dari genggamannya. Masih banyak permasalahan yang harus diatasi seperti rekognisi jenis-jenis dari traffic signs atau rambu lalu lintas.

Traffic Sign atau rambu lalu lintas adalah bagian dari perlengkapan jalan yang memuat lambang, huruf, dan angka. Fungsi dari rambu lalu lintas sangat penting bagi para pengendara karena rambu lalu lintas dapat memberikan informasi yang

krusial, dan terkadang memberi tahu pilihan alternatif yang menarik dalam memilih jalan. Tanpa rambu lalu lintas, pengendara akan cenderung lebih sering terkena kecelakaan lalu lintas karena tidak adanya panduan seperti berapa kecepatan yang aman untuk melaju, atau akan adanya belokan tajam, persimpangan, dan zona sekolah di depannya. Di zaman modern seperti sekarang, setidaknya lebih dari satu juta orang meninggal karena kecelakaan lalu lintas setiap tahunnya. Tentu saja angka tersebut akan meningkat pesat jika rambu lalu lintas tidak ada.

Seperti namanya, autonomous vehicle harus bisa melakukan segalanya secara otomatis dengan intersepsi seminimal mungkin oleh manusia. Maka diharapkan mobil-mobil otomatis ini akan dapat mengklasifikasi dan membedakan rambu lalu lintas yang berada di jalan lalu memproses informasi tersebut menjadi salah satu dari banyak variabel yang digunakan untuk mengambil keputusan. Umumnya, kita menggunakan metode standart computer vision untuk mendeteksi dan mengklasifikasi rambu lalu lintas. Tetapi dibutuhkan waktu yang cukup lama dan sedikit merepotkan karena dikerjakan secara manual. Tetapi apabila kita mengaplikasikan proses deep learning ke dalam masalah ini, kita dapat membuat model yang bisa mengklasifikasi jenis-jenis dari rambu lalu lintas yang dapat diandalkan.

A. Tujuan

Tujuan dari sistem pengenalan rambu lalu lintas adalah dengan adanya sistem ini diharapkan dapat memudahkan kendaraan mengenali rambu lalu lintas pada saat mengendarai mobil.

B. Rumusan Masalah

- Mengetahui tingkat akurasi dari sistem pengenalan rambu lalu lintas menggunakan *Convolutional Neural Network*

II. DASAR TEORI

A. CNN

Convolutional Neural Network adalah metode Machine Learning digunakan untuk mengolah data dua dimensi. CNN masuk ke dalam jenis Deep Neural Network karena memiliki tingkat jaringan yang dalam dan banyak diimplementasikan dalam data citra.

CNN memiliki cara kerja yang hampir serupa dengan *Multilayer Perceptron* (MLP) tetapi neuron di dalam CNN direpresentasikan dalam bentuk dua dimensi, sedangkan dalam MLP hanya ukurannya hanya satu dimensi. Karena data yang dipropagasikan dalam jaringan CNN adalah dua dimensi, maka operasi linear dan parameter bobot pada CNN berbeda. Operasi linear pada CNN menggunakan operasi konvolusi, dan bobotnya berbentuk empat dimensi.

CNN terdiri dari empat *layer*. Pertama adalah *Convolution layer*, yang melakukan operasi konvolusi pada output dari *layer* sebelumnya. Konvolusi sendiri berarti mengaplikasikan fungsi pada output dari fungsi lain secara berulang-ulang. Fungsi dari konvolusi ini adalah untuk mengekstrak fitur dari citra input sehingga menghasilkan transformasi linear dari input sesuai dengan informasi pada data.

Yang kedua adalah *Subsampling layer*, yang merupakan proses reduksi ukuran dari data. *Subsampling* juga digunakan untuk meningkatkan invariansi posisi dari fitur. Lalu yang terakhir ada *Fully connected layer*, biasanya digunakan untuk melakukan transformasi pada dimensi data agar bisa diklasifikasikan secara linear.

III. PENGOLAHAN DATA

Pada penelitian kami Dataset yang kami gunakan sebanyak 12.000 gambar dan untuk mengklasifikasi menggunakan GUI. Kami menggunakan tensorflow dan keras untuk melakukan proses CNN. Pada dataset terdapat beberapa folder yaitu folder test dan folder train. Folder test adalah folder yang berisikan gambar untuk melakukan pengujian. Folder train adalah sebuah folder untuk pembelajaran machine learning kami. Hal pertama yang kami lakukan adalah mengumpulkan seluruh gambar yang sama ke dalam folder dan membuatnya menjadi satu kelas. Langkah kedua yang kami lakukan adalah membuat label untuk mengetahui path dari folder tersebut. Langkah ketiga yang kami lakukan adalah mengubah ukuran gambar menjadi 30 * 30 piksel dan mengubah warnanya menjadi RGB. Langkah keempat kami membuat CNN model.

```
#Retrieving the images and their labels
j=0
for i in range(classes):
    if i == 0 :
        j = 9
    elif i == 1:
        j = 14
    elif i == 2:
        j = 16
    elif i == 3:
        j = 18
    elif i == 4 :
        j = 21
    elif i == 5 :
        j = 27
    elif i == 6:
        j = 28
    elif i == 7:
        j = 34
    elif i == 8:
        j = 35
    elif i == 9:
        j = 38
    path = os.path.join(cur_path, 'Train', str(j))
    images = os.listdir(path)

    for a in images:
        try:
            image = Image.open(path + '\\' + a)
            image = image.resize((30,30))
            image = np.array(image)
            data.append(image)
            labels.append(i)
        except:
            print("Error")
```

CNN yang kami buat menggunakan metode konvolusi dengan aktivasi menggunakan relu. Setelah menggunakan metode konvolusi dan relu maka kami menggunakan polling untuk membuat layer. Kami mengulangi langkah ini sebanyak tiga kali supaya memberikan hasil yang lebih seragam pada contoh untuk training. Setelah melakukan proses diatas kami melakukan training dan validasi model yang sudah kami buat. Kami menggunakan 10.000 sampel dan validasi menggunakan 2.000 contoh gambar sebanyak 15 kali percobaan. Dari hasil yang kami buat tingkat akurasi model kami mencapai 95%.

```
#Building the model
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_train.shape[1:]))
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(10, activation='softmax'))

#Compilation of the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Langkah kelima yang kami lakukan adalah melakukan testing pada model yang sudah kami buat. Cara kita melakukan testing adalah dengan memasukkan gambar kemudian model yang sudah di training mengecek gambar tersebut dan memvalidasinya. Kami membuat GUI supaya memudahkan proses testing. Output dari hasil klasifikasi berupa tulisan.

Know Your Traffic Sign

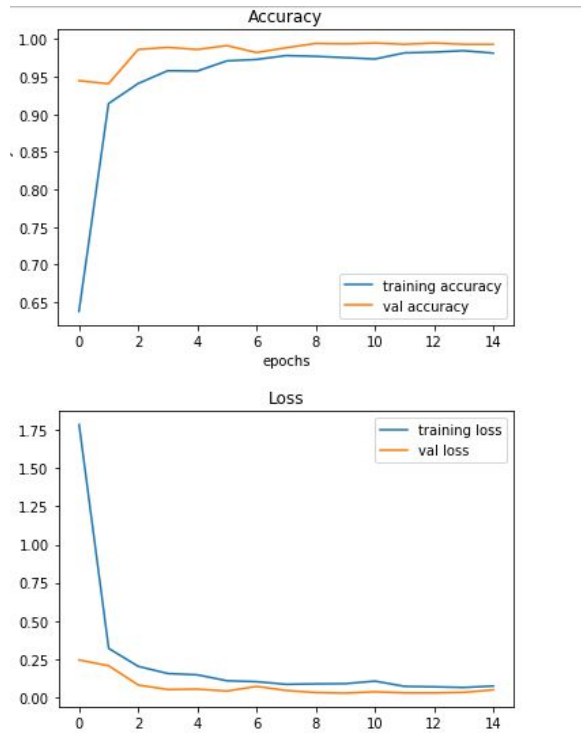
Wajib dilewati

Classify Image



Upload an image

(gambar GUI)



(TABEL TINGKAT AKURASI)

IV. KESIMPULAN

Klasifikasi rambu lalu lintas menggunakan *Convolutional Neural Network* (CNN) dapat dikatakan cukup akurat dengan bukti tingkat akurasi berada pada angka 93%. Klasifikasi dapat dilakukan dengan baik dan juga output yang keluar sesuai hasil yang di inginkan.

DAFTAR PUSTAKA

<https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>

<https://data-flair.training/blogs/python-project-traffic-signs-recognition/>

<https://drive.google.com/drive/folders/1AbLnp2rw18IWTcNKvVF6Zn0V2REuKZw?usp=sharing>