DETEKSI PLAT NOMOR KENDARAAN INDONESIA DENGAN ALGORITMA R-CNN BERBASIS TENSORFLOW DAN EASYOCR

Matthew Brandon Dani[1)] Gregorius Agung Nugroho[1)] Rizky Adrian Pradana[1)] Nabila Husna Shabrina[2)]
[1]Mahasiswa Teknik Komputer Fakultas Teknik dan Informatika Universitas Multimedia Nusantara
[2]Dosen Teknik Komputer Fakultas Teknik dan Informatika Universitas Multimedia Nusantara
[*]Penulis korespondensi: nabila.husna@umn.ac.id

ABSTRAK

Semakin meningkatnya angka penggunaan dan kepemilikan kendaraan bermotor memunculkan masalah baru yaitu meningkatnya angka pencurian kendaraan bermotor. Diperlukan sistem yang dapat memonitor kendaraan yang keluar dan masuk dalam suatu area. Sistem automasi komputer yang dapat mencatat plat nomor kendaraan akan meningkatkan efisiensi sistem tersebut. Oleh karena itu penelitian ini diadakan dengan tujuan untuk meningkatkan keamanan lingkungan sekitar secara efisien dengan menggunakan sistem *deep learning* dengan algoritma R-CNN dan EasyOCR untuk mendeteksi dan pencatatan plat nomor kendaraan beroda empat. Selain itu sistem automasi yang dibuat harus dapat diimplementasikan di perangkat apa saja termasuk yang memiliki kapasitas komputasi yang terbatas. Dengan menggunakan *pre-trained* model VGG-16 dan modifikasi dengan *activation* Softmax, *optimizer* Adam, dan *loss* categorical_crossentropy. Dengan *dataset* sebanyak 611 gambar dan 39 epoch didapatkan model dengan akurasi sebesar 99% dan loss sebesar 0.0246 dalam mendeteksi plat nomor mobil. Dengan melakukan modifikasi algoritma prediksi, waktu yang diperlukan untuk model memprediksi suatu foto kendaraan adalah 10 – 15 detik. Dibuktikan bahwa model *deep learning* dengan algoritma R-CNN berhasil mendeteksi lokasi dari plat nomor dari suatu gambar dan dengan *framework* EasyOCR dapat mencatat plat nomor kendaraan ke dalam *database*.

Kata kunci: R-CNN, EasyOCR, *Machine Learning, Deep Learning*, VGG-16, Deteksi Plat Nomor

ABSTRACT

The increasing number of use and ownership of motor vehicles raises a new problem, namely the increasing number of motor vehicle theft. A system is needed that can monitor vehicles that exit and enter into an area. Computer automation systems that can record vehicle number plates will increase the efficiency of the system. Therefore this research was conducted with the aim of increasing the safety of the surrounding environment efficiently by using a deep learning system with R-CNN and Easyocr algorithms to detect and record four -wheeled vehicle number plates. In addition, the automation system created must be implemented in any device including those that have limited computing capacity. By using the pre-trained model VGG-16 and modification with Activation SoftMax, Adam Optimizer, and Loss Categorical_Crossentropy. With a dataset of 611 images and 39 epochs obtained a model with an accuracy of 99% and a loss of 0.0246 in detecting the car number plate. By modifying the prediction algorithm, the time required for the model predicts a vehicle photo is 10-15 seconds. It is proven that the Deep Learning model with the R-CNN algorithm succeeded in detecting the location of the number plate of an image and with an Easyocr framework can record the vehicle number plate into the database.

Keywords: R-CNN, EasyOCR, Machine Learning, Deep Learning, VGG-16, Plate Number Detection

Pendahuluan

Pencurian kendaraan menjadi kekhawatiran banyak orang. Di Indonesia sendiri, terdapat banyak tempat parkir yang tidak aman atau pengendara yang lupa mengunci kendaraan mereka sehingga mudah dicuri. Menurut BAPPEDA Yogyakarta, pada tahun 2022 terdapat sebanyak 183 kasus pencurian kendaraan bermotor dalam bentuk apa pun (BAPPEDA Yogyakarta, 2023), yang meningkat dari tahun ke tahun. Pencurian kendaraan bermotor terutama mobil membuat masyarakat khawatir terutama mereka yang tinggal di kompleks perumahan atau apartemen yang notabene merupakan tempat dimana orang dapat masuk kapan pun dan terkadang ada orang yang tak dikenal dapat masuk dengan berpura pura sebagai tamu. Sebuah kompleks perumahan ataupun apartemen haruslah memiliki sebuah sistem yang dapat mencegah atau mengurangi kasus pencurian kendaraan bermotor khususnya mobil. Hal ini dapat dilakukan dengan mencatat plat nomor kendaraan yang masuk dan keluar kompleks perumahan atau apartemen sehingga kita bisa mengetahui kendaraan ini kapan ia masuk dan keluar dan apa plat nomornya dan siapa pemiliknya kemudian dicocokkan pada data yang sudah ada sehingga jika ada kasus pencurian dapat ditindak secara langsung. Namun hal ini tidak efisien jika dilakukan secara manual.

Di era serba digital ini, keamanan terhadap pencurian kendaraan bermotor dapat dikurangi dengan berbagai teknologi yang tersedia terutama pada kompleks perumahan dan apartemen dimana tempat masuk dan keluar hanya ada satu pos pemeriksaan saja. Contoh teknologi yang dapat digunakan dalam mengatasi permasalahan tersebut adalah dengan *image processing* dan *deep learning*. *Image processing* digunakan untuk memproses gambar dari hasil tangkapan kamera yang ada di pos penjaga. Gambar atau video dari kamera tersebut nantinya akan di proses menggunakan model yang dikembangkan. Metode *deep learning* digunakan untuk melatih model yang sudah di buat agar model tersebut dapat memproses gambar atau video plat nomor kendaraan yang ditangkap oleh kamera. Plat nomor yang sudah dideteksi nantinya dapat disimpan pada *database* dan dicocokkan saat mobil berada pada pos pemeriksaan yang ditentukan. Idenya adalah, saat ada mobil masuk akan dilakukan pendataan untuk mendata waktu ia masuk dan plat nomornya. Saat keluar, juga dilakukan pendataan waktu dan pencocokan plat nomor dari plat nomor yang ada di *database* jika milik penghuni, jika mobil milik tamu, maka menyerahkan nota / karcis yang diberikan pada saat memasuki kompleks. Dan juga melakukan pencatatan data atau *logging* pada setiap aktivitas keluar masuk kendaraan.

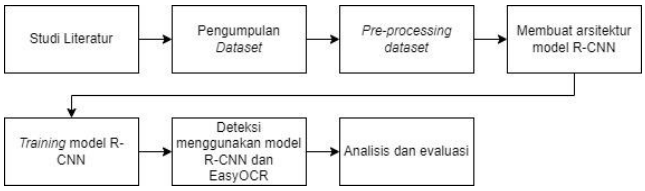
Oleh karena itu, kelompok kami mengembangkan sebuah sistem untuk mendeteksi plat nomor kendaraan dimana kami mengambil *input* gambar dari kamera pada portal menggunakan *image processing* untuk mendeteksi tulisan dari

plat nomor sebuah mobil yang akan masuk ke suatu area seperti parkiran, apartemen, kompleks perumahaan, dan lain lain.

Sistem yang kami buat memiliki target untuk mencapai nilai akurasi di atas 90% dan juga ringan untuk dijalankan di berbagai perangkat yang memiliki kapasitas komputasi yang terbatas. Oleh karena itu kami membuat sistem *image processingnya* dan *deep learning* menggunakan R-CNN dengan *framework* Tensorflow untuk *object detection* nya, dan untuk pembacaan tulisan di fotonya menggunakan *framework* EasyOCR. Untuk mendapatkan model R-CNN kami menggunakan *pre-trained* model VGG-16 yang dimodifikasi sebagai dasar arsitektur. Kami menggunakan dataset berupa foto foto kendaraan yang berasal dari Indonesia dengan mengambilnya melalui website jual beli kendaraan online. Dataset ini akan menjadi data untuk men-training model R-CNN kami. Sehingga sistem yang dibuat dikhususkan hanya untuk kendaraan berplat nomor Indonesia yaitu berwarna hitam dengan tulisan putih.

Metode Penelitian

Terdapat beberapa langkah penelitian yang telah dilakukan oleh penulis untuk mendapatkan hasil penelitian. Yang diperlihatkan pada gambar 1, sebagai berikut :



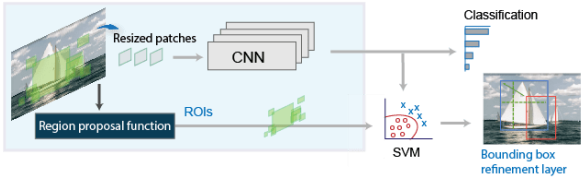
Gambar 1. Alur Penelitian

1. Studi Literatur

Pada tahap ini kami mencari beberapa literasi dan penelitian terdahulu yang berhubungan dengan topik penelitian untuk menentukan arsitektur sistem yang akan dibuat.

1.1.Pengertian R-CNN, Fast R-CNN, dan Faster R-CNN

R-CNN (*Region-based Convolutional Neural Network*) merupakan algoritma untuk mendeteksi sebuah objek yang ada pada sebuah gambar. R-CNN ini terdiri dari dua tahap yaitu: dengan menggunakan *selective search*, R-CNN ini awalnya mengidentifikasi banyak kandidat *bounding box* region dari sebuah objek dan kemudian mengekstrak fitur CNN (*Convolutional Neural Network*) dari tiap region objek secara independen untuk klasifikasi (B.S, 2020).



Gambar 2. Cara Kerja Algoritma R-CNN (MatLab, 2021)

Sebagai ilustrasi, pada gambar 2 diatas, terdapat cara kerja dari R-CNN. 1. R-CNN menerima masukan berupa gambar utuh, 2. kemudian R-CNN mengekstrak sekitar 2000 region proposal. 3. lalu region yang diekstrak tersebut akan diproses untuk mengekstrak fitur CNN-nya, 4. setelah fitur CNN di ekstrak, tiap region akan diklasifikasikan dengan SVM *class specific* (Girhick et al, 2015).

Fast RCNN merupakan sebuah peningkatan dari RCNN dalam segi kecepatan pemrosesan. Fast R-CNN ini menerima masukan sebagai gambar keseluruhan dan sebuah set dari *object proposal*. Fast R-CNN pertama memproses keseluruhan gambar dengan sejumlah *convolutional* dan *max pooling layer* untuk menghasilkan *conv feature map* (Girshick, 2015). Lalu untuk setiap *object proposal*, sebuah *region of interest pooling layer* mengekstrak sebuah fitur vektor dari *feature map*. Fast R-CNN lebih cepat karena Fast RCNN tidak harus mengekstrak 2000 region proposal dan operasi konvolusi (CNN) hanya dilakukan satu kali per gambar.

Terakhir ada Faster R-CNN. Berbeda dengan R-CNN dan Fast R-CNN yang menggunakan *selective search segmentation*. Pada Faster RCNN, region proposal dengan RPN ini dilakukan dengan proses eksternal dan region ini akan diteruskan ke Faster R-CNN (Gavrilescu, R., et al, 2018). jadi dengan pada tahap region proposal dilakukan secara eksternal dari jaringan eksternal.

R-CNN generasi pertama merupakan algoritma R-CNN yang paling sederhana dari pada generasi R-CNN berikutnya seperti Fast R-CNN dan Faster R-CNN.

1.2.Performance Evaluation of Pre-Trained CNN Models for Visual Saliency Prediction (Ghariba, B., et al, 2020)

Pada penelitian ini dibandingkan beberapa *pre-trained* model seperti ResNet-50, VGG16, XCeption, InceptionResNet-V2, dan MobileNet-V2 untuk algoritma CNN. Penelitian ini menghasilkan bahwa waktu komputasi cenderung berbanding lurus dengan hasil akurasi model yang tinggi. Dimana untuk menambah kecepatan prediksi R-CNN namun tetap menjaga akurasi yang tinggi dapat digunakan *pre-trained* model VGG16. Dari hasil penelitian ini dinyatakan bahwa VGG16 memiliki waktu komputasi paling cepat dengan ukuran input gambar paling kecil

yaitu 224x224 pixel namun tetap memiliki akurasi diatas 94%. Ini akan membantu model yang dibuat agar bisa diimplementasikan di perangkat apapun yang memiliki kapasitas komputasi terbatas.

1.3.*A Comparison of YOLO and Mask-RCNN for Detecting Cells from Microfluidic Images* (Mehran G. et al, 2018)

Pada penelitian ini dibandingkan algoritma YOLO dan Mask R-CNN untuk mendeteksi sel pada suatu gambar. YOLO adalah algoritma baru untuk *object detection*. YOLO menggunakan proses CNN sebanyak 1 kali, lalu memprediksi *multiple bounding box*, dan menentukan probabilitas kelas untuk setiap kotak yang tersedia pada *image bounding box*. Algoritma YOLO memiliki kecepatan yang sangat cepat jika dibandingkan dengan RCNN sehingga dapat digunakan untuk *realtime detection*. Hasil penelitian ini disimpulkan bahwa dibandingkan algoritma YOLO dan Mask-RCNN menunjukkan bahwa Mask-RCNN memberikan informasi lebih tentang deteksi sel dari pada YOLO. Walaupun pada penelitian ini mendapatkan hasil penelitian bahwa algoritma YOLO lebih akurat namun sensitif terhadap *noise* atau gangguan.

Kami juga mencari informasi lain bahwa algoritma YOLO harus berjalan di *hardware* yang memiliki GPU agar dapat berjalan cepat, jika algoritma YOLO berjalan pada CPU maka harus memiliki kapasistas komputasi yang tinggi (Rath, S. Gupta, V., 2022).

1.4.*Selective Search for Object Recognition* (Uijlings, J.R.R., 2012)

Pada penelitian ini diajukanlah teknik pencarian *object detection* dengan menggunakan *segmentation selective search*. Dimana menggunakan teknik segmentasi yang nantinya akan menjadi *proposed* region sehingga akan meningkatkan efisiensi pencarian *features* pada *deep learning*. *Segmentation selective search* adalah dasar dari algoritma R-CNN dan Faster R-CNN. Pada penelitian ini di hasilkan bahwa *selective search* berhasil mencari obyek lebih cepat 13 – 59 kali, stabil, dan tahan pada kondisi apa saja.

Dari tahap studi literatur yang sudah dilakukan, kami memutuskan pada penelitian ini menggunakan tensorflow untuk *object detection* dengan algoritma R-CNN dan menggunakan EasyOCR untuk melakukan OCR untuk mendeteksi plat nomor pada kendaraan mobil.

Alasan kami menggunakan Algoritma R-CNN untuk melatih model adalah karena kami menggunakan masukan sistem berupa foto bukan video / *realtime* sehingga dengan menggunakan R-CNN akan lebih akurat dan sederhana ketimbang menggunakan algoritma lainnya seperti Fast R-CNN, Faster R-CNN, atau YOLO. Terlebih R-CNN akan lebih mudah untuk memodifikasinya agar lebih ringan komputasinya dan dapat diimplementasikan pada perangkat apa saja termasuk yang tidak memiliki

GPU. Walau algoritma R-CNN lebih lambat dibanding Faster RCNN, kami lebih mementingkan akurasi dari model dibanding dengan kecepatan karena kami menggunakannya untuk mendeteksi plat nomor yang berhubungan dengan keamanan sehingga dibutuhkan akurasi yang tinggi dan komputasi yang efisien dan sederhana dalam arsitektur. Oleh karena itu kami juga memakai *pre-trained* model VGG16 yang memiliki akurasi yang tinggi namun tidak terlalu berat. *Pre-trained* model VGG16 juga nanti akan kami lakukan modifikasi. Selain itu beberapa penelitian lain sudah menggunakan algoritma YOLO dan Faster R-CNN yang membuat kami ingin mencoba untuk mendeteksi plat nomor kendaraan menggunakan algoritma yang belum pernah diteliti.

Alasan kami menggunakan EasyOCR dibanding dengan OCR yang lain seperti Tesseract adalah karena EasyOCR merupakan OCR yang ringan untuk dijalankan dan pengimplementasian dari EasyOCR ini cukup sederhana sehingga mempersingkat waktu dalam pembuatan model. Dan menurut sumber yang ada, EasyOCR bagus dalam mendeteksi angka yang menurut kami cocok untuk diimplementasikan pada sistem kami (Vedhaviyassh D.R., et al, 2022)

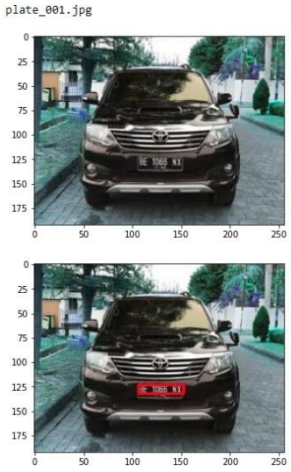
2. Dataset

Pada literatur yang kami temukan, penelitian terdahulu menggunakan *dataset* yang berjumlah besar. Dimana menggunakan sekitar 90.000 gambar pada *dataset*-nya. Kami akan mencoba jika dengan *dataset* yang lebih kecil dari penelitian tersebut apakah tetap mencapai akurasi yang diinginkan. *Dataset* yang kami gunakan merupakan gambar mobil dengan plat dari depan berjumlah 611 gambar yang dibagi menjadi 80% untuk *training* dan 20% untuk testing. Sumber gambar ini kami dapatkan dari situs jual beli mobil *online* di internet. Berikut merupakan beberapa contoh dataset yang kami gunakan yang dapat dilihat pada gambar 3:

Gambar 3. Contoh Dataset

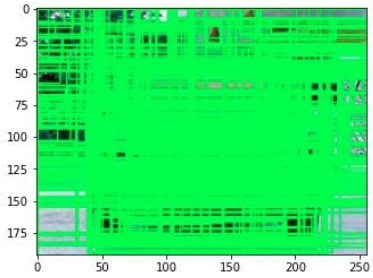
Sebelum dimasukan gambar tersebut harus dilakukan *labelling* data sehingga dapat digunakan untuk *training*. kami menggunakan perangkat lunak bernama LabelImg untuk melakukan labelling dimana outputnya berbentuk

file Pascal VOC atau XML. Untuk *training* akan menggunakan *file* berbentuk CSV, oleh karena itu harus diubah terlebih dahulu bentuk *file*-nya. Berikut contoh dataset yang diberi label region pada gambar 4:



Gambar 4. Dataset Region

Setelah itu menggunakan CV2.selective search segmentation dimana akan dibagi bagi per region untuk nantinya digunakan hasil training. Untuk contoh dari selective search segmentation dapat dilihat pada gambar 5:



Gambar 5. Cara kerja Selective Search Segmentation

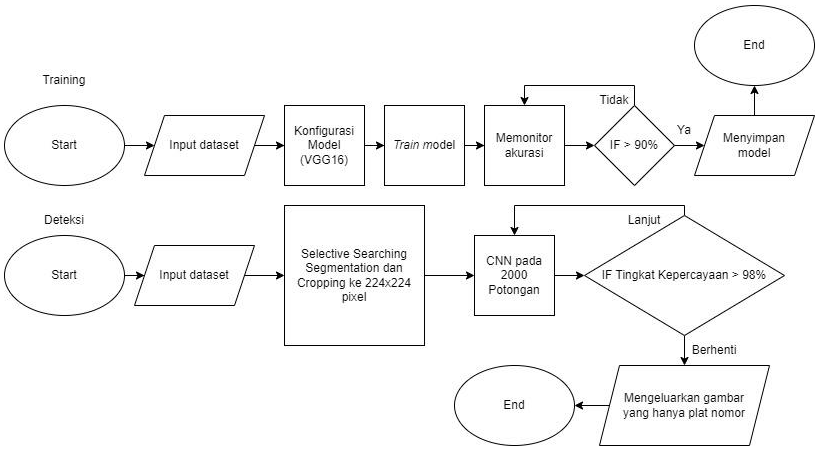
Dimana tidak semua segmentasi dijadikan data, namun segmentasi yang memiliki nilai *Intersection over Union* (IoU) d iatas 70% dan di bawah 30% sajalah yang dipakai untuk *false* dan *true* dataset. IoU adalah sebuah metrik yang digunakan untuk memperkirakan suatu perpotongan mendekati kebenaran atau tidak. Setelah dataset dilakukan selective search segmentation, jumlah total dataset berjumlah 18557 potongan dan setelah segmentasi dari gambar dataset akan di resize ke ukuran 224 x 224 pixel agar memiliki ukuran yang sama serta membuat beban untuk training model lebih ringan. Sehingga untuk jumlah akhir dataset adalah 14925 potong untuk dataset training dan 3732 potong untuk dataset testing dan validation, dataset ini memiliki 2 kategori label yaitu *true* atau *false*

Berikut merupakan flowchart dari proses pengumpulan dataset kami yang dapat dilihat pada gambar 6.

Gambar 6. Flowchart pengumpulan dataset

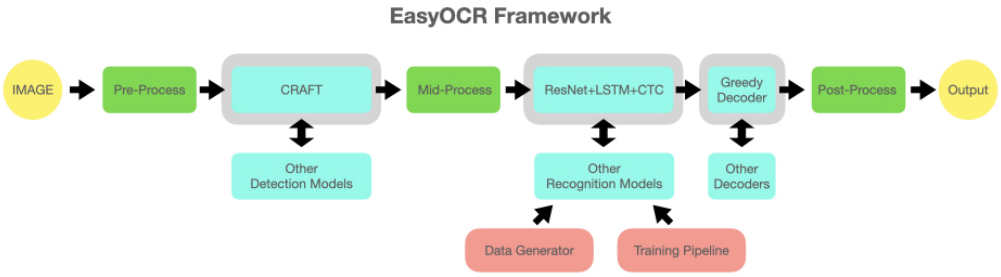
3. R-CNN & EasyOCR

Proyek yang kami kerjakan dibagi ke dalam dua tahap yaitu tahap *object detection* dan tahap *optical character recognition* (OCR). Pada tahap *object detection*, masukannya merupakan gambar jpg dan keluarannya merupakan gambar yang sudah ter-*crop* hanya pada plat nomor. Pertama, gambar JPG akan diubah menjadi format yang bisa diproses lebih lanjut. Tensorflow berperan dalam mendeteksi plat nomor yang ada pada gambar mobil lalu gambar tersebut di proses dengan algoritma R-CNN sehingga model dapat mengenali sebuah plat pada mobil yang ada pada gambar. Sesudah mendeteksi plat, kemudian gambar akan di-*crop* sehingga yang terlihat hanya plat nomornya saja. Model yang dikerjakan akan dilatih secara terus menerus sehingga sebisa mungkin menyentuh angka di atas 90 %. Untuk alur algoritma object detection dapat dilihat pada gambar 7.



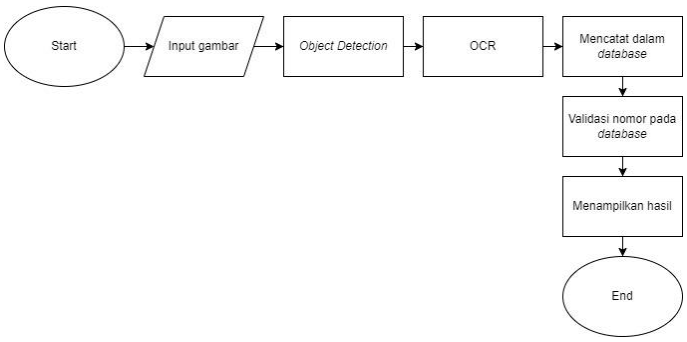
Gambar 7. Flowchart cara kerja object detection

Kemudian tahap selanjutnya adalah tahap OCR. Pada tahap ini, masukannya yang diterima adalah gambar yang sudah diproses pada tahap *object detection* dan keluarannya adalah text. gambar plat nomor yang sudah ter-*crop* sebelumnya akan diproses oleh OCR yang akan mengekstrak teks pada gambar menjadi tulisan yang nantinya akan disimpan pada *file* CSV. Alur kerja algoritma EasyOCR dapat dilihat pada gambar 8.



Gambar 8. Flowchart cara kerja OCR (JaidedAI, 2020)

Kami akan membuat aplikasi akhir dengan library Tkinter untuk mencontohkan implementasi penelitian ini, alur kerja aplikasi dapat dilihat pada gambar 9.



Gambar 9. Flowchart cara kerja aplikasi akhir

Hasil dan Pembahasan

Pada proyek ini kami menggunakan library Tensorflow Keras versi 2.7 untuk *framework training* model, EasyOCR versi 1.4.1 untuk melakukan proses *Optical Character Recognition*, dan karena data yang kami gunakan berformat csv, maka dibutuhkan *pre-processing* dengan *library* Pandas. Untuk *interfacing* sebagai contoh program akhir kami menggunakan *library* Tkinter. Spesifikasi perangkat yang kami gunakan untuk *training* dan testing adalah :

Processor : Intel Core I7-4720HQ (arsitektur x64)

RAM : 12GB

Model R-CNN kami menggunakan basis dari arsitektur VGG16 yang merupakan sebuah model *pre-trained* yang sudah dibuat sebelumnya. Kami menambahkan *output layer* pada modelnya berupa *dense layer* dengan besar 2 karena untuk kategori pada datasetnya ada 2 yaitu *true* dan *false*. Model kemudian di *compile* dengan *activation* softmax, *optimizer* adam dan *loss* categorical_crossentropy. Loss berupa categorical crossentropy karena label pada dataset kami berupa binary yaitu 0 (*false*) atau 1 (*true*). Hasil arsitektur model *summary* dijabarkan sebagai berikut pada gambar 10:

Model: "model"

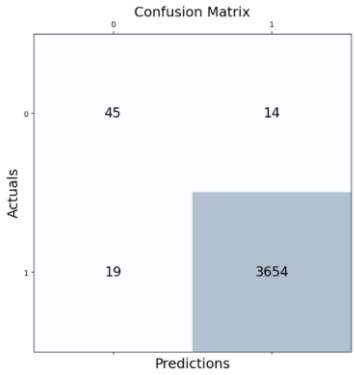
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
dense (Dense)	(None, 2)	8194

Total params: 134,268,738
Trainable params: 126,633,474
Non-trainable params: 7,635,264

Gambar 10. Hasil Arsitektur Modul R-CNN

Parameter untuk model.fit dan train model adalah : steps_per_epoch = 10, epochs = 100 , validation_steps = 2, dan menggunakan fungsi *callback* berupa earlyStopping. Fungsi *callback* ini adalah untuk memakai model dari epoch yang memiliki *validation loss* lebih rendah daripada epoch sebelumnya. Contoh : pada epoch 1 memiliki vall_loss sebesar 1 dan epoch 2 memiliki val_loss sebesar 3, maka model pada epoch 2 tidak akan di save karena memiliki val_loss lebih besar daripada epoch 1. Kami mengejar nilai *loss* karena walaupun akurasi tinggi namun jika ada prediksi yang salah tidak terlalu jauh dari nilai sebenarnya.

Waktu yang dibutuhkan untuk *training* 2 jam, dan waktu *training* per epoch nya adalah sekitar 1 menit atau 78 detik. Model yang kami dapat berada pada epoch ke 39 karena setelah epoch tersebut, val_loss dan akurasi tidak meningkat lagi dan untuk menghindari model yang *overfitting*. Hasil yang kami dapat dari epoch tersebut adalah akurasi sebesar 99.06 %, *loss* sebesar 0.0246, *validation loss* sebesar 0.0001, dan *validation accuracy* sebesar 1. Dan untuk *Confusion Matrix* model dengan memprediksi 3732 potong dari *dataset test* dapat dilihat pada gambar 11:



Gambar 11. *Confusion Matrix* dari model yang didapat

Dapat dilihat dari gambar 11, bahwa sebagian besar potongan berhasil di prediksi secara benar hanya 33 potong atau 0.88% saja yang diprediksi salah.

Setelah di *training*, model tersebut digunakan untuk mendeteksi plat nomor dari gambar. Perangkat yang digunakan sama seperti perangkat pada *training*, Gambar plat akan terdeteksi setelah 15 detik. hasilnya dapat dilihat pada gambar 12 untuk prediksi yang benar dan salah.

Gambar 12. Hasil deteksi dan OCR Gambar (Kiri Benar, Kanan Salah)

Untuk deteksi RCNN secara penuh atau benar benar mendeteksi dari seluruh 2000 region membutuhkan waktu sekitar 3 menit dengan perangkat yang sama. Kami bisa mendapatkan waktu 15 detik karena saat suatu region dideteksi benar dengan tingkat kepercayaan lebih dari 98% maka program akan segera memberikan hasil akhir dan berlanjut pada proses berikutnya. Hal ini merupakan kelebihan menggunakan algoritma R-CNN sehingga bisa memodifikasi algoritma prediksi sehingga memunculkan hasil lebih cepat.

Hasil deteksi plat yang didapat terbilang cukup akurat karena dari gambar keseluruhan, hanya plat mobilnya saja yang terdeteksi secara sempurna. Namun untuk hasil OCR dari EasyOCR-nya sendiri tidak sesuai dengan tulisan yang ada pada plat mobil dan ada huruf atau angka yang seharusnya tidak ada pada foto plat nomor. Hal ini seperti deteksi baut penempel plat mobil yang dideteksi sebagai titik, lalu format plat nomor yang tidak selalu sama dimana ada yang memiliki spasi ada juga yang tidak. Lalu jika hasil OCR sudah ada, maka akan dicocokkan pada plat yang sudah ada pada *database* csv. Jika terdaftar, maka akan muncul tulisan plat terdaftar. Jika plat tidak ada di *database*, maka akan muncul tulisan plat tidak terdaftar.

Kesimpulan

Berdasarkan hasil penelitian yang didapat, model R-CNN yang kami buat sudah bisa mendeteksi plat nomor secara akurat dengan tingkat akurasi model sebesar 99.06%, *loss* sebesar 0.0246, *validation loss* sebesar 0.0001, dan *validation accuracy* sebesar 1. Waktu untuk mendeteksi suatu foto kendaraan bermotor dan OCR tulisan pada plat kendaraan yaitu sekitar 10 – 15 detik. Algoritma prediksi juga dapat dijalankan di berbagai perangkat yang tidak memiliki GPU karena prediksi menggunakan komputasi CPU. Hasil akhir penelitian berupa aplikasi yang dapat mencatat plat nomor kendaraan beserta waktu saat dilakukan prediksi ke dalam *database* berupa file .csv dan memvalidasi apakah plat nomor tersebut terdaftar pada *database*. Oleh karena itu hasil penelitian ini memenuhi tujuan awal penelitian ini dibuat.

Dan juga hanya dengan 600 gambar dataset sudah memiliki akurasi sebesar 99% yang berarti sistem ini memiliki kemungkinan bahwa akan mudah diimplementasikan di kasus lain seperti plat nomor kendaraan negara lain.

Namun terdapat satu masalah yaitu untuk sistem OCR-nya tidak begitu memuaskan karena masih ada huruf atau angka yang tidak terdeteksi ataupun huruf atau angka yang seharusnya tidak ada namun terdeteksi pada OCR. Hipotesis penulis adalah pada beberapa plat kendaraan bermotor terdapat baut untuk menempelkan plat ke kendaraan, hal ini menjadi gangguan bagi sistem OCR yang digunakan. Dan juga kemungkinan penyebab hasil OCR yang tidak begitu baik adalah ketidakcocokan *framework* easyOCR dengan kasus plat nomor

kendaraan Indonesia, hasil model yang dibuat yaitu dengan resolusi gambar kurang dari 224 x 224 pixel, ataupun *framework* yang digunakan kurang *powerful* untuk mendeteksi huruf dan angka.

Hasil aplikasi penelitian ini dapat diimplementasikan di berbagai pos penjagaan tempat parkir seperti di tempat parkir apartemen, rumah sakit, pusat pembelanjaan, kantor, dan lain lain. Aplikasi ini juga dapat diimplementasikan menjadi aplikasi web dengan mengimplementasikannya pada suatu *backend* layanan web dengan *framework* Flask, FastAPI, dan lain lain, sehingga bisa digunakan di perangkat genggam untuk E-Tilang, dan sebagainya.

Berdasarkan Perpol Nomor 7 Tahun 2021 Pasal 45 Ayat 1 (a) (Korlantas Porli, 2021), bahwa pada kendaraan bermotor milik perseorangan dan beberapa pihak lainnya akan menggunakan plat nomor berwarna putih dengan tulisan hitam. Penelitian ini tetap bisa diimplementasikan pada *dataset* plat nomor berwarna putih dengan tulisan hitam dengan cara menambahkan metode *inverting* warna pada saat *pre-processing* gambar yang akan diprediksi. Dimana akan membalikkan warna ke berlawanannya seperti dari intensitas cahaya putih menjadi hitam. Dan untuk plat nomor berwarna lain, bisa dilakuka *pre-processing* data yaitu melakukan pengurangan *channel* warna menjadi 1 *channel* yaitu hanya hitam atau putih.

Untuk penelitian kedepannya, kami merencanakan penggunaan framework atau library OCR yang lain selain easyOCR untuk meningkatkan hasil akurasi dari OCR agar huruf dan angka pada plat bisa sesuai dengan yang aslinya. Seperti menggunakan framework Tesseract namun dengan dataset sendiri.

Ucapan Terima Kasih

Puji syukur kepada tuhan Yang Maha Esa atas berkat yang diberikan dalam pelaksanaan dan pembuatan artikel ilmiah dengan judul “Deteksi Plat Nomor Kendaraan Indonesia dengan Algoritma R-CNN berbasis Tensorflow dan EasyOCR” sehingga dapat terselesaikan. Kami juga ucapkan banyak terima kasih kepada Bu Nabila selaku dosen mata kuliah Image Processing (CE649-AL) karena sudah mengajarkan banyak ilmu yang berguna untuk keberhasilan penelitian ini dan memberikan motivasi dan semangat. Semoga artikel ilmiah ini dapat bermanfaat sebagai sumber informasi dan inspirasi bagi para pembaca. Penulis juga berharap dapat melanjutkan pengembangan lebih lanjut untuk artikel ilmiah ini dan penelitian yang lain agar dapat aktif memajukan pendidikan dan literasi di Indonesia.

Kontribusi Penulis

Dalam penelitian yang telah dilakukan, Matthew Brandon Dani memiliki peran untuk melakukan *pre-processing dataset* yang telah dikumpulkan, membuat

algoritma *deep learning* R-CNN dan EasyOCR, dan menganalisis hasil model penelitian. Gregorius Agung Nugroho memiliki peran untuk mengumpulkan *dataset* dan sumber referensi dan membuat aplikasi untuk memprediksi menggunakan *library* tkinter. Rizky Adrian Pradana memiliki peran untuk mengumpulkan *dataset*, membantu Analisa hasil model penelitian, dan membuat algoritma *database*. Ketiga penulis juga bersama sama melakukan studi pustaka, pengujian, dan pembuatan artikel ilmiah ini. Ibu Nabila Husna Shabrina sebagai dosen pembimbing memiliki peran untuk mendampingi dan memvalidasi hasil penelitian yang telah dilakukan dan saat penyusunan artikel ilmiah ini.

Daftar Pustaka

B. S., Rekha.. 2020. Object Detection using Region based Convolutional Neural Network: A Survey. *International Journal for Research in Applied Science and Engineering Technology.* 8(7):1927–1932.

BAPPEDA Yogyakarta. 2023. Jumlah Kasus Pencurian. URL: http://bappeda.jogjaprov.go.id/dataku/data_dasar/index/447-jumlah-kasus-pencurian?id_skpd=147. Diakses tanggal 20 Februari 2023.

Gavrilescu, R., Zet, C., Fosalau, C., Skoczylas, M., & Cotovanu, D. 2018. Faster R-CNN:an Approach to Real-Time Object Detection. *2018 International Conference and Exposition on Electrical And Power Engineering (EPE).* 2018, Iasi, Romania. pp. 0165-0168.

Ghafari, M., Mailman, D., Hatami, P., et al. 2022. A Comparison of YOLO and Mask-RCNN for Detectiong Cells from Microfluidic Images. *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. 2022, Jeju Island, Republic of Korea. pp. 204-209.

Ghariba, B., Shehata, M. S., McGuire, P,. 2020. Performance Evaluation of Pre-Trained CNN Models for Visual Saliency Prediction. *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2020, London, ON, Canada. pp. 1-4.

Girshick, R. 2015. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision, 2015 International Conference on Computer Vision.* ICCV 2015: 1440–1448.

Girshick, R., Donahue, J., Member, S., & Darrell, T. 2015. *Region-based Convolutional Networks for Accurate Object Detection and Segmentation*. 8828(c): 1–16.

JaidedAI. 2020. EasyOCR. URL: https://github.com/JaidedAI/EasyOCR. Diakses tanggal 25 November 2021.

Korlantas Polri. 2021. Peraturan Kepolisian Negara Republik Indonesia tentang Registrasi dan Identifikasi Kendaraan Bermotor. URL: https://korlantas.polri.go.id/wp-content/uploads/2021/05/PERATURAN-POLRI-NOMOR-7-TAHUN-2021-TTG-REGIDENT-RANMOR-TGL-5-MEI-2021.pdf. Diakses tanggal 1 Desember 2021.

Matlab. 2021. Getting Started with R-CNN, Fast R-CNN, and Faster R-CNN. URL: https://www.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html. Diakses tanggal 25 November 2021.

Rath, S., Gupta, V. 2022. Performance Comparison of YOLO Object Detection Models – An Intensive Study. URL: https://learnopencv.com/performance-comparison-of-yolo-models/. Diakses tanggal 24 Februari 2023.

Uijlings, J.R.R., Sande, K.E.A, Gevers, T., Smeulders, A.W.M. 2012. Selective Search for Object Recognition. URL: http://www.huppelen.nl/publications/selectiveSearchDraft.pdf. Diakses tanggal 27 November 2021

Vedhaviyassh, D., R., Sudhan R., Saranya, G., et al. 2022. Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm. *2022 6th International Conference on Electronics, Communication and Aerospace Technology.* 2022, Coimbatore, India. pp. 966-97.