

Indonesian Number Plate Detection Using R-CNN Algorithm with Tensorflow and EasyOCR

Matthew Brandon Dani¹, Tezard Almafazan Mulia², William Putra Pratama Wijaya³

^{1,2,3,4} Teknik Komputer, Universitas Multimedia Nusantara, Tangerang, Indonesia

¹ matthew.brandon@student.umn.ac.id, ² tezard.alfafazan@student.umn.ac.id, ³ william.putra@student.umn.ac.id

Diterima 26 November 2021

Disetujui dd mmmmm yyyy

Abstract—Penelitian ini diadakan dengan tujuan untuk meningkatkan keamanan lingkungan sekitar dengan menggunakan sistem deep learning RCNN dan EasyOCR untuk mendeteksi dan pencatatan plat nomor kendaraan beroda empat. Dengan menggunakan pretrained 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% dalam mendeteksi plat nomor mobil. Dibuktikan bahwa model berhasil mendeteksi lokasi dari plat nomor dari suatu gambar.

Index Terms—RCNN, EasyOCR, Deep Learning, VGG-16, Deteksi Plat

I. INTRODUCTION

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 BAPPENAS, pada tahun 2013 terdapat sebanyak 42.508 kasus pencurian kendaraan bermotor dalam bentuk apapun [1], yang meningkat dari tahun ketahun. 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 kapanpun 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 checkpoint saja. Contoh teknologi yang dapat digunakan dalam mengatasi permasalahan tersebut adalah dengan *image processing* dan *deep learning*. *Image processing* digunakan untuk menangkap gambar atau video dari kendaraan oleh kamera yang terpasang pada *checkpoint*. 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 checkpoint 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 dari gambar dari kamera pada portal menggunakan *image processing* untuk mendeteksi tulisan dari plat nomor sebuah mobil yang akan masuk ke parkir.

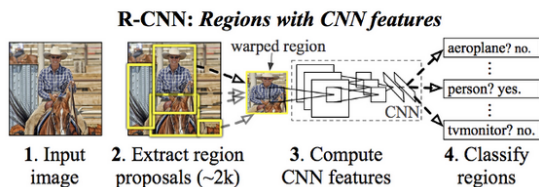
Sistem yang kami buat memiliki target untuk mencapai nilai akurasi diatas 90% dan juga ringan untuk dijalankan di berbagai device. Oleh karena itu kami membuat sistem *image processing*nya dengan sistem deep learning menggunakan RCNN dengan framework Tensorflow untuk *object detection* nya, dan untuk pembacaan text di imagenya menggunakan *framework* EasyOCR. Untuk mendapatkan model

RCNN nya kami menggunakan *pretrained* model VGG-16 yang dimodifikasi sebagai dasar arsitekturnya.

II. LITERATURE REVIEW

Pada penelitian ini kami menggunakan algoritma RCNN untuk mendeteksi plat nomor pada kendaraan mobil.

RCNN (*Region-based Convolutional Neural Network*) merupakan merupakan algoritma untuk mendeteksi sebuah objek yang ada pada sebuah gambar. RCNN ini terdiri dari dua tahap yaitu: dengan menggunakan selective search, RCNN 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 [2].



Gambar 1 : Cara kerja RCNN

Sebagai ilustrasi, pada gambar 1, terdapat cara kerja dari RCNN. 1. RCNN menerima input berupa gambar utuh, 2. kemudian RCNN 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*. [4]

Selain RCNN yang kami gunakan, ada juga algoritma lain yang mirip dengan RCNN ini yaitu *Fast RCNN* dan *Faster RCNN*.

Fast RCNN merupakan sebuah peningkatan dari RCNN dalam segi kecepatan pemrosesan. *Fast RCNN* ini menerima input sebagai gambar keseluruhan dan sebuah set dari *object proposal*. *Fast RCNN* pertama memproses keseluruhan gambar dengan sejumlah *convolutional* dan *max pooling layer* untuk menghasilkan *conv feature map*[3]. Lalu untuk setiap *object proposal*, sebuah *region of interest pooling layer* mengekstrak sebuah fitur vektor dari *feature map*. *Fast RCNN* lebih cepat karena *Fast RCNN* tidak harus mengekstrak 2000 *region proposal* dan operasi konvolusi (CNN) nya hanya dilakukan satu kali per gambar.

Terakhir ada *Faster RCNN*. Berbeda dengan RCNN dan *Fast RCNN* yang menggunakan *selective search segmentation* . Pada *faster RCNN*, *region proposal* dengan RPN ini dilakukan dengan proses

eksternal dan region ini akan diteruskan ke *faster RCNN*[5]. jadi dengan pada tahap *region proposal* dilakukan secara eksternal dari network eksternal.

Pada literatur yang kami temukan, penelitian penelitian tersebut menggunakan dataset yang berjumlah besar. Dimana menggunakan sekitar 90.000 gambar pada datasetnya. Kami akan mencoba jika dengan dataset yang lebih kecil dari penelitian tersebut apakah tetap mencapai akurasi yang diinginkan.

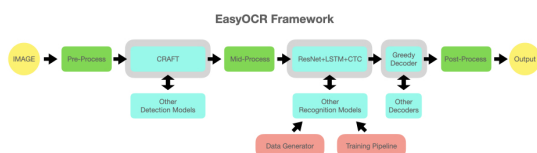
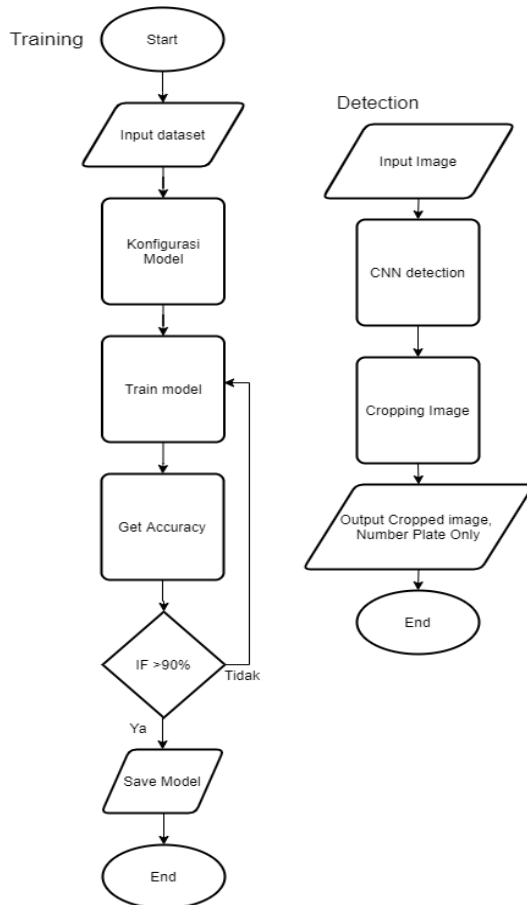
III. METODOLOGI DAN IMPLEMENTASI PROYEK

Proyek yang kami kerjakan dibagi kedalam dua tahap yaitu tahap *object detection* dan tahap *optical character recognition* (OCR). Pada tahap *object detection*, inputnya merupakan gambar jpg dan outputnya 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 Region based Convolutional Neural Network (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 nomor saya. Kemudian tahap selanjutnya adalah tahap *Optical Character Recognition* (OCR). Pada tahap ini, input yang diterima adalah gambar yang sudah diproses pada tahap *detection* dan outputnya adalah text. gambar plat nomor yang sudah ter crop sebelumnya akan diproses oleh OCR yang akan mengekstrak teks pada gambar menjadi text yang nantinya akan disimpan pada file CSV. Model yang dikerjakan akan dilatih secara terus menerus sehingga sebisa mungkin menyentuh angka diatas 90 %.

Pada project ini sudah ditulis sebelumnya bahwa kelompok kami akan menggunakan tensorflow untuk object detection dengan algoritma R-CNN dan menggunakan EasyOCR untuk melakukan OCR. Alasan kami menggunakan Algoritma R-CNN untuk melatih model adalah karena kami menggunakan gambar berupa foto dibandingkan dengan video / real time sehingga dengan menggunakan R-CNN akan lebih akurat ketimbang menggunakan algoritma lainnya seperti *Fast R-CNN* atau *Faster R-CNN*. 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. 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.

Flowchart dari proyek yang kami kerjakan serta *flowchart* dari framework EasyOCR yang digunakan dapat dilihat dalam gambar berikut :

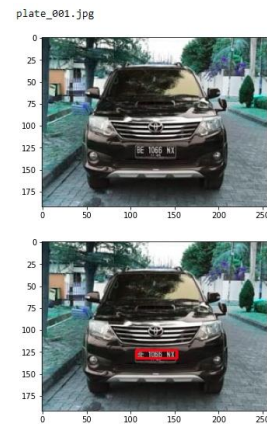


Gambar 2 : Flowchart sistem yang dikembangkan dan easyOCR

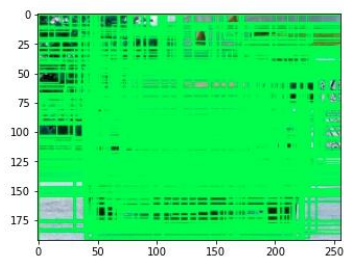
IV. DATASET

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. Sebelum diinput gambar tersebut harus dilakukan labelling

data sehingga dapat digunakan untuk *training*. kami menggunakan *software* bernama Labellmg 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 filenya.



Setelah itu menggunakan *CV2.selective search segmentation* dimana akan dibagi bagi per region untuk nantinya digunakan hasil *training*. Untuk contoh dari *selective search segmentation* seperti berikut.



Dimana tidak semua segmentasi dijadikan data, namun segmentasi yang memiliki nilai *Intersection over Union (IoU)* diatas 70% dan dibawah 30% sajalah yang dipakai untuk *false* dan *true* dataset. *IoU* adalah sebuah metrik yang digunakan untuk memperkirakan suatu perpotongan mendekati kebenaran atau tidak. Setelah itu 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.

V. Hasil dan Analisis

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 *preprocessing* dengan library *pandas*. Untuk interfacing sebagai contoh program akhir kami menggunakan library *tkinter*. Spesifikasi laptop yang kami gunakan untuk *training* dan *testing* adalah :

Processor : Intel I7-4720HQ

RAM : 12GB
GPU : Nvidia GTX 950M

Model R-CNN kami menggunakan basis dari arsitektur VGG16 yang merupakan sebuah model *pretrained* yang sudah dibuat sebelumnya. Model kemudian di compile dengan *activation softmax*, *optimizer adam* dan *loss categorical_crossentropy*. dengan model summary sebagai berikut :

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)	590880
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590880
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		

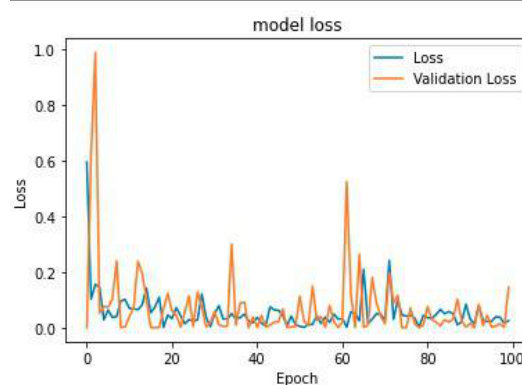
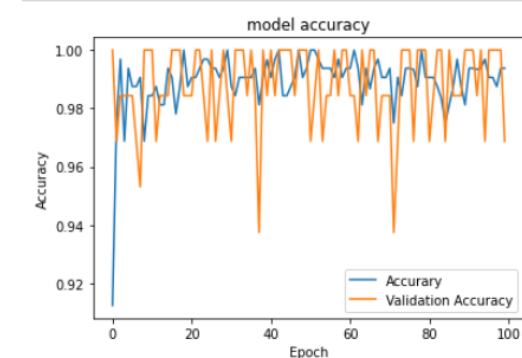
Gambar 3 Hasil Model Summary

Parameter untuk model.fit dan train model adalah : *steps_per_epoch* = 10, *epochs* = 100 , *validation_steps* = 2, dan menggunakan *callback*. Fungsi *callback* ini adalah untuk memakai model dari epoch yang memiliki val_loss lebih rendah daripada epoch sebelumnya. Contoh : pada epoch 1 memiliki val_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.

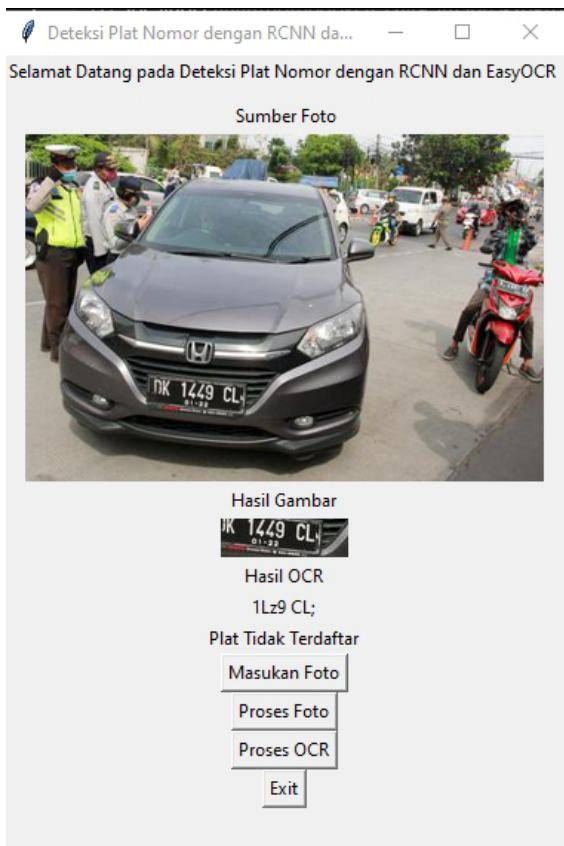
Waktu yang dibutuhkan untuk training 2 jam, dan *time* per epoch nya adalah sekitar 1 menit atau 78 detik.

Model yang kami dapat berada pada epoch ke 39 karena setelah epoch tersebut, val_loss dan akurasi tidak meningkat lagi. 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. Sedangkan model pada epoch ke 100 memiliki akurasi sebesar 99.37 %, akurasi validation sebesar 0.9688, loss sebesar 0.0260 dan validation loss sebesar 0.1456. Hasil ini dapat dilihat dalam graph berikut :



Gambar 4 Grafik kurva akurasi model dan model loss

Setelah di *training*, model tersebut digunakan untuk mendeteksi plat nomor dari gambar. *Hardware* yang digunakan sama seperti *hardware* pada *training*. Gambar plat akan terdeteksi setelah 15 detik. hasilnya adalah sebagai berikut :



Gambar 5 Hasil deteksi dan OCR

Untuk deteksi RCNN secara penuh atau benar benar mendeteksi dari seluruh 2000 region

membutuhkan waktu sekitar 3 menit dengan hardware 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.

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. Lalu jika hasil OCR sudah ada, maka akan dicocokkan pada plat yang sudah ada pada database csv. Jika terdaftar, maka akan muncul plat terdaftar. Jika plat tidak ada di database, maka akan muncul plat tidak terdaftar.

VI. KESIMPULAN

Berdasarkan hasil yang didapat, model yang kami buat sudah bisa mendeteksi plat nomor secara akurat dengan tingkat akurasi model sebesar 99% namun 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. Kemungkinan hasil OCR yang tidak begitu baik adalah ketidakcocokan framework easyOCR dengan model yang kami buat ataupun framework yang digunakan kurang *powerful*.

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

Kami ucapkan banyak terimakasih 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.

DAFTAR PUSTAKA

- [1] (BPS, 2014)BPS. (2014). Kependudukan Badan Pusat Statistik. *Statistic Indonesia*. bps.go.id: <http://www.bps.go.id/subject/12/kependudukan.html>
- [2] (B. S, 2020)B. S, R. (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. <https://doi.org/10.22214/ijraset.2020.30430>
- [3] (Girshick, 2015)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. <https://doi.org/10.1109/ICCV.2015.169>

- [4] (Girshick et al., 2015) Girshick, R., Donahue, J., Member, S., & Darrell, T. (2015). *Region-based Convolutional Networks for Accurate Object Detection and Segmentation*. 8828(c), 1–16. <https://doi.org/10.1109/TPAMI.2015.2437384>
- [5] (Lokanath et al., 2017) Lokanath, M., Kumar, K. S., & Keerthi, E. S. (2017). Accurate object classification and

detection by faster-RCNN. *IOP Conference Series: Materials Science and Engineering*, 263(5). <https://doi.org/10.1088/1757-899X/263/5/052028>