

# Deteksi Senjata Genggam Menggunakan Faster R-CNN Inception V2

Ivandi Christiani Pradana, Eko Mulyanto, dan Reza Fuad Rachmadi  
Departemen Teknik Komputer, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: ekomulyanto@ee.its.ac.id

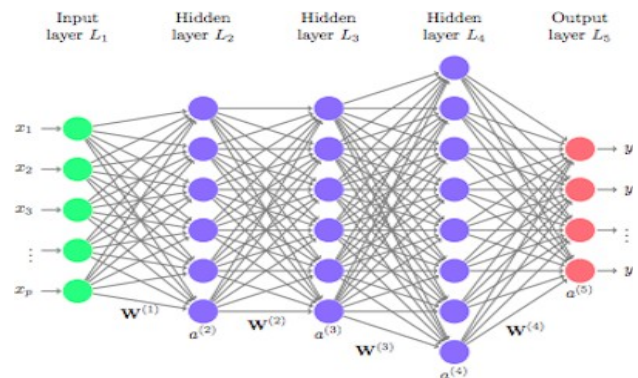
**Abstrak**—Senjata genggam kelas pisau sering digunakan dalam kegiatan kriminal di Indonesia. Sering kali objek pisau yang terekam sulit terlihat dengan mata telanjang. Proses deteksi senjata genggam kelas pisau bisa dibantu dengan pengolahan citra menggunakan Deep Learning. Penelitian ini bertujuan untuk mengaplikasikan konsep Deep Learning dan Tensorflow Object Detection untuk melatih model Faster R-CNN Inception V2 untuk bisa mendeteksi senjata genggam kelas pisau dalam citra digital. Dalam penelitian ini, model yang terlatih bisa menandai benda yang diduga sebagai senjata genggam kelas pisau dalam gambar dengan kotak penanda. Model yang dibuat dari penelitian ini dilatih dengan kumpulan dataset berisi gambar senjata genggam kelas pisau, dataset didapat dan terdiri dari rekaman bela diri pisau dan kumpulan pisau dengan bentuk dan warna yang beragam. Penelitian ini meneliti akurasi model Faster R-CNN Inception V2 yang dilatih dalam mendeteksi senjata genggam kelas pisau. Hasil akhir dari proses pengembangan model Faster R-CNN Inception V2 ini adalah model yang berhasil mendeteksi senjata genggam kelas pisau dengan akurasi sebanyak 87%, hasil akurasi didapatkan dari pengujian terhadap 475 gambar digital yang dilakukan di Google Colab.

**Kata Kunci**—Inception-V2, Deep Learning, TensorFlow, Pisau.

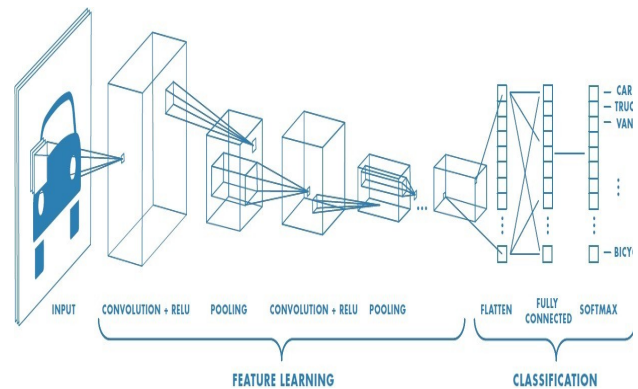
## I. PENDAHULUAN

TINDAKAN kriminal terus terjadi di setiap negara, salah satunya Indonesia. Salah satu kegiatan kriminal yang marak diberitakan dan berpotensi mengancam nyawa adalah begal. Begal adalah tindakan kriminal mencuri kendaraan bermotor (khususnya motor) tindakan begal di Indonesia kebanyakan menggunakan senjata genggam untuk mengintimidasi korban supaya mereka menyerahkan kendaraan mereka [1]. Cctv merupakan alat yang sering dipakai untuk memantau untuk mencegah atau menjadi barang bukti ketika kegiatan kriminal terjadi [2]. Namun kualitas gambar cctv yang cenderung buruk beserta ukuran senjata yang cenderung kecil membuat barang bukti seperti senjata genggam sulit ditemukan atau diidentifikasi dengan mata telanjang [3].

Penelitian bidang visi komputer mencoba untuk mendeteksi ada tidaknya senjata genggam dalam sebuah gambar dan juga mengenali jenis, macam, dan variasi senjata genggam tersebut. Namun dari riset terkait kebanyakan masih mengarah untuk mendeteksi senjata api seperti pistol dan mengandalkan lingkungan yang terang dikarenakan memiliki bahaya yang lebih signifikan dan menghasilkan pembacaan lebih akurat [4]. Penelitian terkait pendeteksi senjata genggam tentunya sudah dilakukan beberapa lembaga dengan tujuan dan hasil yang bervariasi. Salah satu penelitian ini merupakan riset untuk mengembangkan model pendeteksi senjata genggam otomatis berbasis CNN dengan menambahkan preprocessing berbasis penerangan citra bernama DaCoLT (Darkening and Contrast at Learning and Test stages). Pendirian dataset perangkat ini menggunakan dataset dari 3 kategori untuk senjata, latar belakang, dan objek mirip senjata secara berurut dengan total

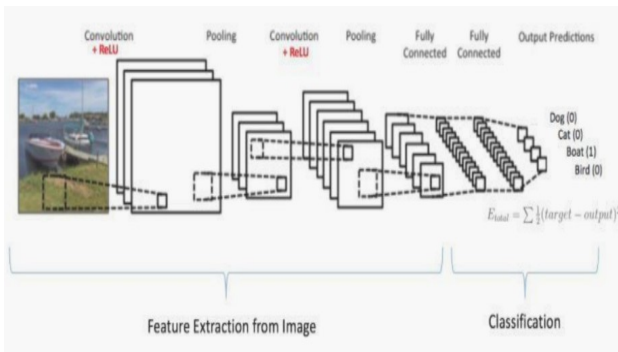


Gambar 1. Deep neural network.

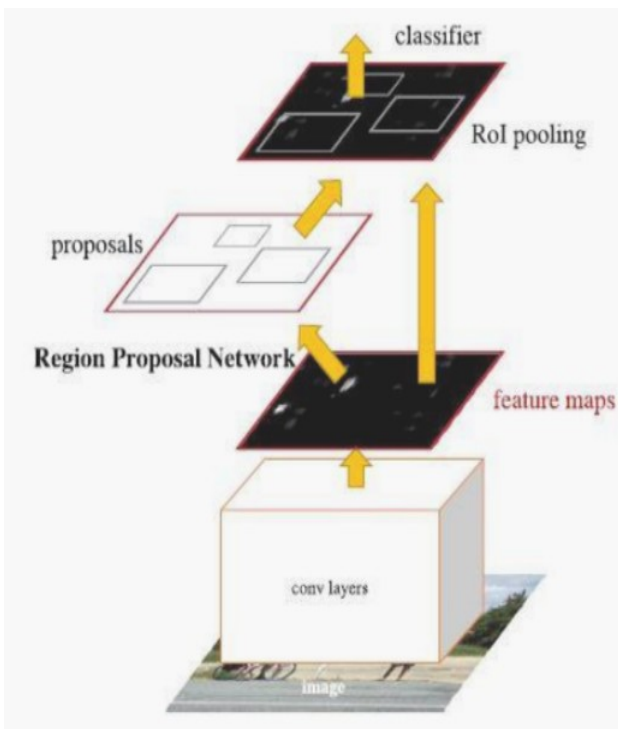


Gambar 2. Visualisasi arsitektur CNN.

semua dataset terdiri dari sekitar 16000 gambar. Untuk pelatihan digunakan percobaan dengan menggunakan 3 model yakni SSD, R-FCN, Faster R-CNN dimana kondisi masing-masing diperiksa akurasi berdasarkan penerangan objek yang berbeda-beda. Percobaan dengan menggunakan Dacolt menghasilkan hasil yang menjanjikan dengan meningkatkan akurasi deteksi senjata dalam lingkungan indoor [5]. Percobaan selanjutnya merupakan pendeteksi senjata berbasis CNN untuk mendeteksi senjata api. Dataset merupakan dataset senjata api yang kebanyakan diambil dari IMFDB dimana senjata api yang digunakan merupakan pistol, revolver, senapan, dan shotgun. Dataset kedua digunakan untuk menunjukkan hasil negatif atau objek background. Model diimplementasikan menggunakan toolbox Matlab MatConvNet menggunakan model pre trained VGG-16. Percobaan melakukan uji coba akurasi berdasarkan algoritme dengan hasil akurasi rata-rata 93 persen dibanding dengan riset lain [6]. Percobaan ketiga merupakan pendeteksi senjata berbasis CNN menggunakan algoritme Faster R-CNN dan SSD (Single SHot Detection). Dataset berisi senjata dengan background dimana senjata ditandai dengan memberi label di masing-masing gambar, 80 persen digunakan untuk pelatihan dan 20 persen dari dataset digunakan untuk uji coba akurasi. Hasil yang didapat model menggunakan SSD memiliki akurasi 79 persen yang lebih rendah dibanding hasil menggunakan model Faster R-CNN yang memiliki akurasi 98 persen, namun hasil ini sendiri masih bergantung dengan



Gambar 3. Proses CNN secara keseluruhan.



Gambar 4. Faster R-CNN.

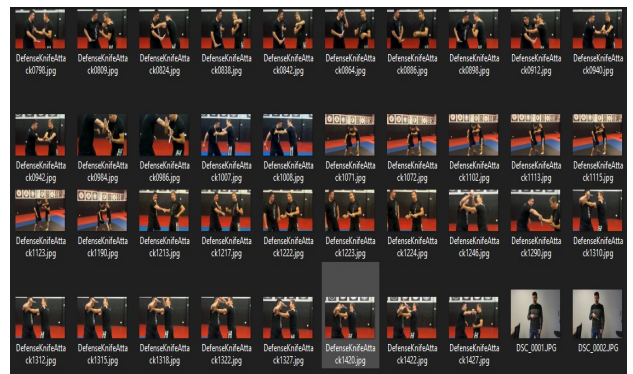
senjata yang dideteksi dimana AK-47 cenderung dideteksi dengan akurasi lebih tinggi oleh model dibanding dengan senjata api lain utamanya pistol mirip Colt M1911 [7].

Muncul insentif untuk membuat perangkat yang dapat mendeteksi senjata genggam kelas pisau yang merupakan kelas yang paling bisa diakses oleh penduduk Indonesia. Untuk proyek ini, model deep learning untuk mendirikan perangkat deteksi senjata tajam ini adalah *Faster R-CNN Inception V2*.

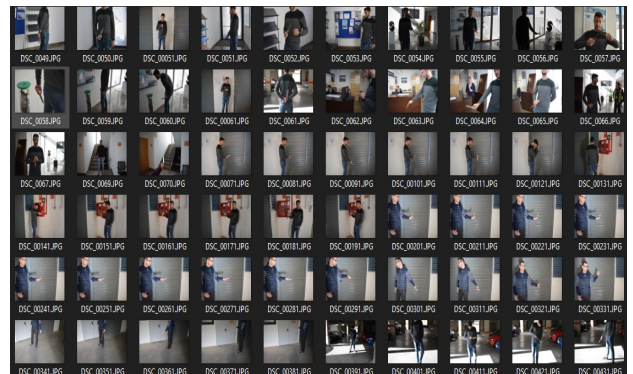
Berdasarkan data yang telah dipaparkan di latar belakang dapat dirumuskan permasalahan untuk tugas akhir ini, proses untuk menemukan senjata genggam kelas pisau dalam gambar digital susah dilakukan dengan mata telanjang.

## II. DESAIN DAN IMPLEMENTASI

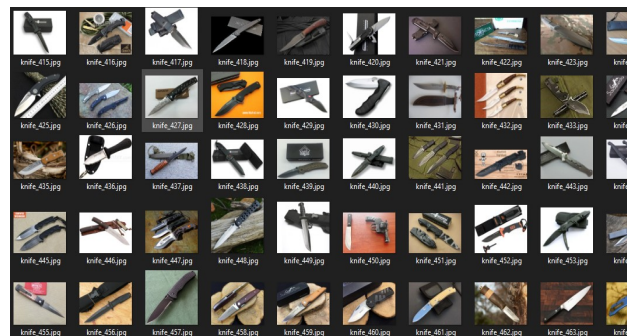
Tugas akhir ini secara umum bekerja untuk menerapkan pengetahuan *Deep Learning* dengan mengumpulkan gambar dan pra-pemrosesan gambar. *Deep Learning* adalah bagian dari keluarga metode *Machine Learning* yang lebih luas berdasarkan *neural network* dengan pembelajaran representasi. Mereka terdiri dari sejumlah besar node yang terhubung, yang masing-masing melakukan operasi matematika sederhana. Output setiap node ditentukan oleh operasi ini, serta seperangkat parameter yang spesifik untuk node tersebut. Dengan menghubungkan node ini bersama-sama dan dengan



Gambar 5. Dataset dari video.



Gambar 6. Dataset dari dasci.

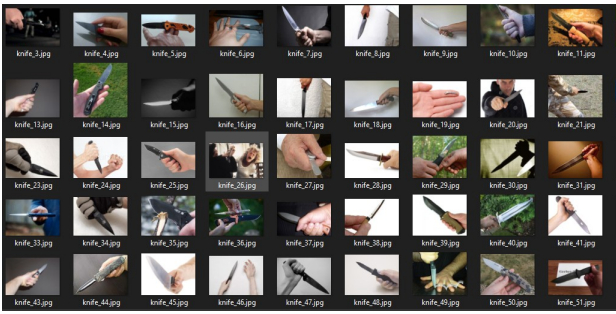


Gambar 7. Dataset objek.

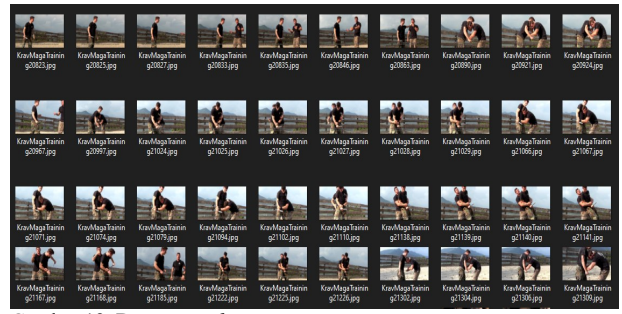
hati-hati mengatur parameternya, fungsi yang sangat kompleks dapat dipelajari dan dihitung seperti yang dirujuk oleh gambar 1. *Neural Network* memiliki berbagai perbedaan dari otak biologis. Secara khusus, jaringan saraf tiruan cenderung statis dan simbolis, sedangkan otak biologis sebagian besar organisme hidup bersifat dinamis (plastik) dan analog.. Selanjutnya dataset tersebut dibagi menjadi dataset pelatit dan dataset penguji, dimana dataset pelatit digunakan untuk melatih model. Setelah model dilatih, dataset penguji akan digunakan untuk menguji model untuk melakukan proses evaluasi. Model terlatih ini dapat digunakan untuk menerima masukan dari input gambar statik dengan format .jpg yang disimpan di Google Drive. Setelah menghubungkan program ke data gambar yang terdapat di basis data Google Drive, program dapat mengeluarkan keluaran berupa segmentasi area senjata genggam kelas pisau.

### A. CNN Architecture

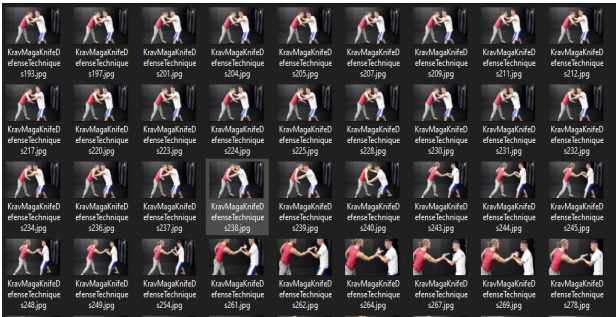
CNN merupakan salah satu jenis *Neural Network* yang biasa digunakan untuk mengolah data dua dimensi. CNN digunakan untuk mendeteksi dan mengenali suatu objek dalam sebuah citra. Secara garis besar CNN hampir sama dengan *neural network* biasa. Hal yang membedakannya terletak pada arsitekturnya. CNN memiliki proses untuk mengekstrak fitur



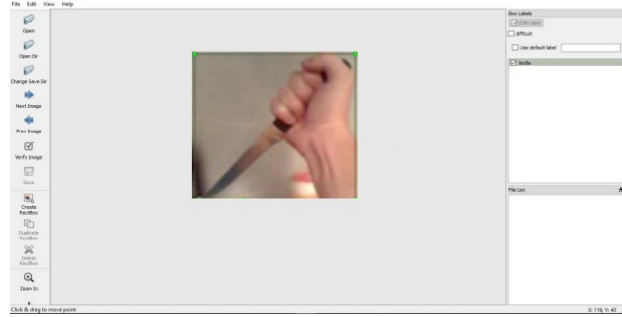
Gambar 8. Dataset pisau yang digenggam.



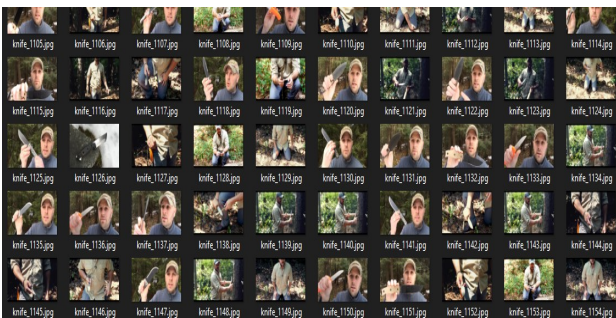
Gambar 12. Dataset outdoor tereng.



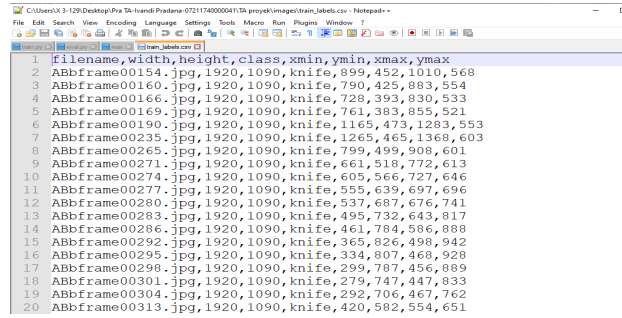
Gambar 9. Dataset bela diri.



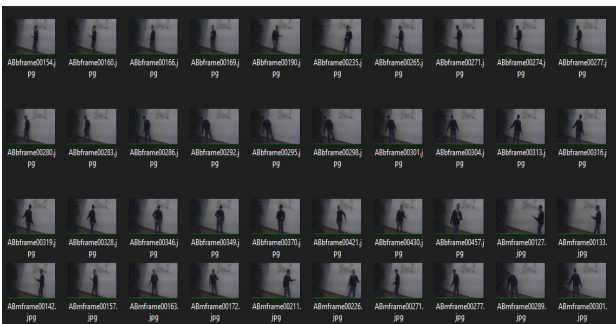
Gambar 13. Demonstrasi labellmg.



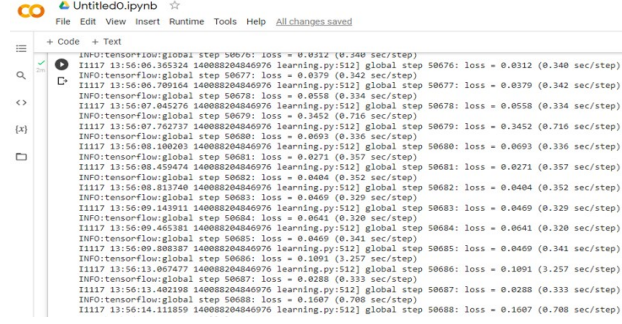
Gambar 10. Dataset close up.



Gambar 14. Contoh file .csv.



Gambar 11. Dataset pencahayaan redup.



Gambar 15. Loss dari proses pelatihan.

dari sebuah gambar. Prosesnya terdiri dari dua bagian yaitu Convolutional Layer dan Pooling Layer seperti terlihat pada Gambar 1.

B. Pendirian Dataset

Data yang digunakan dalam tugas akhir ini adalah kumpulan dataset senjata tajam yang sebagian besar diperoleh dari Andalusia Research Institute in Data Science and Computational Intelligence (DaSci) (Gambar 2, Gambar 3, dan Gambar 4). Selain menggunakan dataset dari DaSci, dataset lainnya juga dibuat dari footage CCTV yang memiliki kualitas gambar senjata dalam skala yang lebih kecil dan memiliki kualitas yang relatif rendah.

Sehingga data tersebut dapat diolah melalui proses pelatihan dengan CNN. Data tersebut harus diolah terlebih dahulu dengan cara mengekstraknya menjadi data citra per frame agar dapat dilakukan tahap preprocessing. Data citra yang telah

diekstrak dari video harus masuk ke tahap pre-processing sebelum melakukan proses pelatihan. Preprocessing ini adalah augmentasi data. tujuan dari augmentasi data adalah untuk meningkatkan variasi dari dataset. Augmentasi yang dilakukan pada data adalah: (1) Random Zoom, (2) Random Rotate.

Proses preprocessing dilakukan secara manual sehingga tidak terjadi di semua gambar. Gambar-gambar dataset diambil dari rekaman berbagai macam sudut dan konteks pada gambar 5, dimana juga dima- sukkan gambar senjata genggam itu sendiri dan dalam posisi tidak digenggam pada Gambar 7 dan juga dalam posisi sedang digenggam sehingga terlihat tangan aktor penggenggam pada Gambar 8. Juga ada senjata genggam ya ng sedang digenggam oleh aktor dan direkam dengan sudut peletakan cctv pada umumnya pada Gambar 6, selain dari sudut rekaman cctv pada umumnya juga terdapat dataset peragaan penggunaan senjata genggam kelas pisau

```

Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

11118 16:37:46.307173 140235385235328 eval_util.py:376] Running eval batches done.
INFO:tensorflow: success: 475
11118 16:37:46.307409 140235385235328 eval_util.py:381] # success: 475
INFO:tensorflow: skipped: 0
11118 16:37:46.307567 140235385235328 eval_util.py:382] # skipped: 0
11118 16:37:46.454432 140235385235328 object_detection_evaluation.py:1335] average_precision: 0.870351
INFO:tensorflow:Writing metrics to tf summary.
11118 16:37:46.707987 140235385235328 eval_util.py:90] Writing metrics to tf summary.
INFO:tensorflow:Losses/Loss/BoxClassifierLoss/classification_loss: 0.070293
11118 16:37:46.708301 140235385235328 eval_util.py:97] Losses/Loss/BoxClassifierLoss/classification_loss: 0.040404
INFO:tensorflow:Losses/Loss/BoxClassifierLoss/localization_loss: 0.040404
11118 16:37:46.709024 140235385235328 eval_util.py:97] Losses/Loss/BoxClassifierLoss/localization_loss: 0.018873
INFO:tensorflow:Losses/Loss/RPNLoss/localization_loss: 0.018873
11118 16:37:46.709388 140235385235328 eval_util.py:97] Losses/Loss/RPNLoss/localization_loss: 0.018073
INFO:tensorflow:Losses/Loss/RPNLoss/objectness_loss: 0.069031
11118 16:37:46.709716 140235385235328 eval_util.py:97] Losses/Loss/RPNLoss/objectness_loss: 0.069031
INFO:tensorflow:PascalBoxes_PerformanceByCategory/AP@0.5IOU/knife: 0.870351
11118 16:37:46.710072 140235385235328 eval_util.py:97] PascalBoxes_PerformanceByCategory/AP@0.5IOU/knife: 0.870351
INFO:tensorflow:PascalBoxes_Precision/AP@0.5IOU: 0.870351
11118 16:37:46.710412 140235385235328 eval_util.py:97] PascalBoxes_Precision/AP@0.5IOU: 0.870351
INFO:tensorflow:Metrics written to tf summary.
11118 16:37:46.710694 140235385235328 eval_util.py:98] Metrics written to tf summary.
    
```

Gambar 16. AP hasil pengujian.



Gambar 19. Deteksi gambar rekaman berhasil.



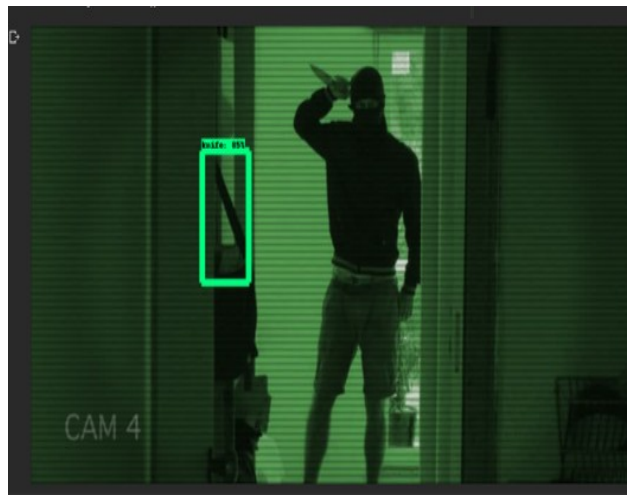
Gambar 17. Hasil close up benar.



Gambar 20. Gagal deteksi.



Gambar 18. Deteksi gambar bayangan berhasil.



Gambar 21. Salah deteksi.

tersebut dalam posisi penyerangan pada Gambar 9, gambar rekaman pisau dalam pencahayaan yang lebih redup dari yang lain pada Gambar 11 dan juga peragaan penggunaan senjata genggam di lingkungan *outdoor*. Gambar senjata genggam kelas pisau yang diletakkan dekat dengan muka pada Gambar 10.

C. Image Labelling

Tahap selanjutnya adalah pelabelan citra untuk klasifikasi objek tertentu pada citra menggunakan labelIm untuk menggambar kotak penanda yang juga memberikan deskripsi

objek di dalam kotak dan disimpan dalam file xml. Kemudian semua file xml akan diubah menjadi satu file csv. Setelah semua gambar diberi label dan csv telah dibuat, proses selanjutnya adalah mengubah dataset pelatit dan pengujian menjadi TFrecords yang merupakan format yang didukung oleh TensorFlow untuk menyimpan gambar ke dalam urutan string biner (Gambar 12).

D. Distribusi Dataset

Dalam pelatihan mesin, adalah umum untuk membagi dataset. Komposisi dapat bervariasi tergantung pada jumlah gambar dalam kumpulan data di mana semakin banyak gambar yang terkandung dalam kumpulan data semakin rendah

persentase data uji, distribusi persentase dataset umumnya sebagai berikut: 80% Perangkat Kereta, dan 20% Perangkat Pengujian dan 70% Set Kereta, dan 30% Set Tes 1) Set Pelatihan (Gambar 13 dan Gambar 14) [8]. Train Set adalah kumpulan data yang digunakan untuk proses pelatihan model. Dalam penelitian ini, persentase total dataset train dibandingkan dengan seluruh dataset adalah 78% [9].

Tes Set digunakan untuk proses pengujian model. Dataset uji terdiri dari 22% dari total dataset dalam penelitian ini

#### E. Training Process

Setelah dataset disiapkan, dataset diupload ke model untuk pelatihan. Tugas akhir ini menggunakan Tensor Flow, Google object detection API dan Google Collaboratory karena keterbatasan hardware. Proses pelatihan dimulai dengan memasang Google object detection API ke TensorFlow di platform Google Colab dengan memasukkan library ke dalam folder Google Drive di Google Co lab dan menginstal library ke TensorFlow. Eksperimen ini menggunakan model yang telah dilatih sebelumnya, yaitu Google Inception-V2 [10]. Model ini dapat diunduh di TensorFlow Github. Model ini sebelumnya telah dilatih menggunakan dataset Common Objects in Context (COCO) sehingga bobot dan bias telah disesuaikan dengan objek di dataset COCO [9]. Konsep penggunaan kembali pengetahuan dan kumpulan data ini disebut *transer learning* untuk mempercepat penelitian dengan menyederhanakan proses pe modelan ulang tanpa perlu melatih lagi menggunakan kumpulan data yang sama persis. Setelah semua dokumen disiapkan, model dapat dilatih dengan memasukkannya ke dalam folder drive Google Colab dan menginisialisasi proses pelatihan dengan file ipynb.

#### F. Testing Process

Setelah pelatihan selesai, proses selanjutnya adalah proses pengujian yang memberikan evaluasi dengan memasukkan dataset pengujian sebagai input. Proses ini berguna untuk menentukan keberhasilan pelatihan model ini.

### III. HASIL DAN PEMBAHASAN

#### A. Dataset Distribution

Distribusi dataset tugas akhir ini menggunakan komposisi 72% dataset sebagai data latih dan sisanya 28% dataset sebagai data uji, menjadikan penelitian ini mendekati persentase pengujian tertinggi karena dataset hanya memiliki 2.136 citra.

#### B. Training and Validation Result

Dalam proses pelatihan, jumlah kerugian untuk setiap langkah global dapat dilihat di file acara di TensorBoard. Jumlah langkah yang diambil sekitar 50000 dengan nilai kerugian tidak kurang dari 0,2. pada Gambar 15.

#### C. Testing Result

Proses pengujian dilakukan dengan menggunakan dataset pengujian. Proses pengujian dilakukan dengan GPU dari Google Colab. Hasil AP dari proses pengujian sebanyak 475 citra untuk objek kelas senjata tajam adalah 0,870351% seperti terlihat pada Gambar 16 dan Gambar 17.

#### D. Inferencing Result

Proses pengujian dengan gambar secara langsung menghasilkan keberhasilan dan kesalahan dimana ditemukan kesalahan pada proses pendeteksian senjata tajam, dimana benda yang bukan senjata tajam (*sling bag*) dianggap senjata tajam

seperti terlihat pada Gambar 18, Gambar 19, dan Gambar 20.

Hasil dari pengujian dari model pelatihan ini juga dibandingkan dengan model sebelumnya. Hasil pengujian model sebelumnya bisa dilihat di Gambar 21. Hasil pengujian model lama menghasilkan AP sebanyak 0.813601 dengan *Classification Loss* sebanyak 0.112305. Jika dibandingkan dengan model yang sudah dilatih terdapat peningkatan sebanyak 0.05675 untuk AP dan 0.042012 untuk *Classification Loss*.

### IV. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, Google Colab bisa digunakan untuk pengujian model *deep learning*. Hasil pembacaan gambar bisa dilakukan dengan pengunggahan gambar secara *online* ke Google Drive sehingga tidak memakan banyak penyimpanan dalam perangkat keras pribadi. Selain itu, model *deep learning* sebelumnya bisa digunakan untuk pelatihan model baru tanpa perlu membuat model dari awal dengan dataset digunakan sebelumnya untuk mendirikan model tersebut.

Proses pelatihan untuk model ini bisa dibilang berlangsung terlalu lama dikarenakan telah terjadi sekitar 50.000 *steps* dan sudah berlangsung sekitar 4 jam. Selain memakan waktu yang lama, durasi pelatihan juga bisa dipangkas karena hasil *loss* sudah konsisten dalam rentang 0.x sekitar 2 jam pelatihan sehingga bisa menghemat waktu dan energi listrik khususnya ketika menggunakan pelatihan secara lokal. Berkaitan alasan penggunaan Google Colab adalah dikarenakan keterbatasan *hardware* dan proses pelatihan ini dilakukan diluar kampus sehingga tidak bisa menggunakan laboratorium kampus untuk mendapat hasil dari *hardware* yang memadai. Melihat dari hasil AP dan mAP yang sama karena hanya terdapat 1 kelas dalam proyek ini, terdapat hasil -87% untuk mendeteksi kelas *knife* yang berarti terjadi peningkatan sebanyak -6% dari model sebelumnya. Meski hasil AP yang didapat cukup memuaskan, tetap terdapat *error* dalam deteksi objek dari gambar yang diunggah untuk proses pengujian langsung diluar dataset pengujian. *Error* ini terjadi di gambar berkualitas redup dan resolusi rendah dimana ada gambar yang tidak mendeteksi pisau dan salah memetakan posisi pisau dan menandakan objek yang bukan pisau sebagai pisau.

Pengembangan lebih lanjut dari penelitian ini bisa dilakukan dengan meningkatkan keakuratan model dengan menambah kelas baru untuk melatih model tersebut. Hal ini bertujuan untuk mencegah terjadinya kesalahan dalam memetakan objek dan mengurangi kasus *false positive* dalam deteksi senjata tajam dalam proyek ini. Dataset juga bisa ditambah dalam jumlah dan variasinya meningkatkan deteksi pisau yang berbentuk unik, untuk proses ini prioritaskan kuantitas gambar untuk tiap variasi model pisau yang akan dilatih untuk mencegah masalah *overfitting*.

### DAFTAR PUSTAKA

- [1] F. Muhammad, "Tinjauan Kriminologis Tentang Kejahatan Begal yang Menggunakan Senjata Tajam (Studi Sasus di Kota Makassar Tahun 2011- 2015)," Universitas Hasanuddin Makassar, 2018.
- [2] N. B. Haq, "Peran Humas Polres Lumajang dalam Meminimalisir Tindak Pidana Begal di Kabupaten Lumajang," Universitas Muhammadiyah Jember, 2018.
- [3] M. Gill and K. Loveday, "What do offenders think about CCTV?," *Crime Prev. Community Saf.*, vol. 5, no. 3, pp. 17-25, 2003, doi: 10.1057/palgrave.cpcs.8140152.
- [4] S. A. A. Shah, M. A. Al-Khasawneh, and M. I. Uddin, "Review of

- Weapon Detection Techniques within the Scope of Street-Crimes,” in *2nd International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, 2021, pp. 26–37, doi: 10.1109/ICSCEE50312.2021.9498007.
- [5] A. Castillo, S. Tabik, F. Pérez, R. Olmos, and F. Herrera, “Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning,” *Neurocomputing*, vol. 330, pp. 151–161, 2019, doi: 10.1016/j.neucom.2018.10.076.
- [6] G. K. Verma and A. Dhillon, “A Handheld Gun Detection Using Faster R-CNN Deep Learning,” in *Proceedings of the 7th international conference on computer and communication technology*, 2017, pp. 84–88, doi: 10.1145/3154979.3154988.
- [7] A. K. Jain, B. Klare, and U. Park, “Face matching and retrieval in forensics applications,” *IEEE Multimed.*, vol. 19, no. 1, p. 20, 2012.
- [8] M. Mourad, J.-L. Bertrand-Krajewski, and G. Chebbo, “Calibration and validation of multiple regression models for stormwater quality prediction: data partitioning, effect of dataset size and characteristics,” *Water Sci. Technol.*, vol. 52, no. 3, pp. 45–52, 2005, doi: 10.2166/wst.2005.0060.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Adv. Neural Inf. Process. Syst.*, vol. 28, no. 1, 2015.
- [10] T. Mustafa, S. Dhavale, and M. M. Kuber, “Performance analysis of inception-v2 and Yolov3-based human activity recognition in videos,” *SN Comput. Sci.*, vol. 1, no. 3, pp. 1–7, 2020, doi: 10.1007/s42979-020-00143-w.