

# Implementasi objek detection dan tracking menggunakan deep learning untuk pengolahan citra digital

Jefry Sunupurwa Asri<sup>1)</sup>, Gerry Firmansyah<sup>2)</sup>

Universitas Esa Unggul

Jl. Arjuna Utara No. 9, Kebon Jeruk, Jakarta Barat 11510

e-mail: jefrysunupurwa@gmail.com<sup>1)</sup>, gerry@esaunggul.ac.id<sup>2)</sup>

## Abstrak

Artikel ini bertujuan untuk mendeskripsikan peningkatan hasil pengolahan citra digital melalui objek detection dan tracking dengan memanfaatkan arsitektur deep learning. Metode yang digunakan dalam penelitian ini adalah metode deep learning, CAMShift, dan Faster R-CNN yang dilalui dengan serangkaian tahapan, yaitu: merancang dan mengimplementasikan metode yang digunakan dengan Raspberry Pi menggunakan Python dan Library OpenCV, mengkombinasikan Faster R-CNN dengan CAMShift, testing proses yang dibangun menggunakan data real time video yang telah dikumpulkan, setelah itu mengevaluasi kinerja dari segi akurasi terhadap masing-masing proses menggunakan hasil pengolahan tahap penggabungan metode yang telah dikumpulkan, dan menarik kesimpulan. Berdasarkan hasil studi penelitian, maka dapat di simpulkan bahwa penggabungan metode tersebut dapat dilakukan, akan tetapi masih memiliki beberapa kelemahan, yaitu pada saat proses penyimpanan data digital sangat banyak membutuhkan waktu proses penyimpanan yang lama, dan akurasi citra gambar digital belum memuaskan.

**Kata kunci:** Object Tracking, Object Detection, Deep Learning, Pengolahan Citra Digital.

## 1. Pendahuluan

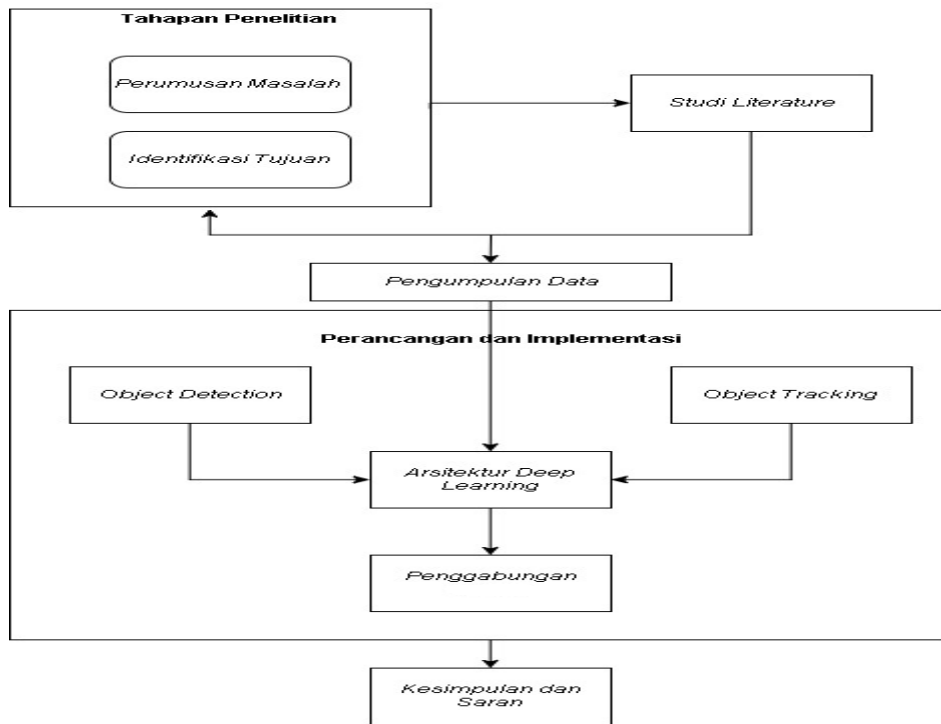
Pada era digital saat ini sudah menjadi kebutuhan untuk memanfaatkan citra digital (gambar atau video) dalam berbagai keperluan. Hal ini dapat dilihat dari peranan citra digital tersebut dalam memberikan informasi tentang apa yang ada dalam citra digital tersebut dengan menggunakan arsitektur deep learning. Deep Learning adalah sebuah bidang keilmuan dalam bidang Machine Learning yang akhir-akhir ini berkembang karena teknologi GPU (Graphics Processing Unit) Acceleration. Deep Learning memiliki kemampuan sangat baik dalam computer vision salah satunya dalam klasifikasi objek pada citra digital.

Objek Detection dan tracking merupakan penelitian yang menarik dibidang computer vision. Objek Tracking adalah langkah pertama dalam berbagai aplikasi yang ada di Indonesia dalam bidang robotik, keamanan, pengawasan, citra medis, olahraga, dll. Objek Tracking biasanya mendeteksi kehadiran benda-benda yang ditangkap oleh kamera, mengunci target, menandai dan mengikuti benda bergerak tersebut. Ada berbagai metode yang sering dipakai dimana masing-masing memiliki kelebihan dan kekurangan. Salah satunya metodenya ialah CAMShift singkatan dari (Continuously Adaptive Mean-Shift). [1],[3]

Objek detection sendiri memiliki peranan dalam menentukan benda-benda yang ditangkap oleh kamera tersebut tersimpan didalam database atau tidak. Sehingga bisa menyimpulkan hasil dari benda yang telah di tracking tadi. Ada berbagai metode yang sering dipakai dimana masing-masing metode memiliki kelebihan dan kekurangannya. Salah satu metodenya ialah Faster R-CNN (Convolutional Neural Network). [7]

Dengan mempertimbangkan penelitian sebelumnya peneliti berkeinginan untuk mencoba menerapkan penggabungan metode tersebut dengan arsitektur deep learning sebagai landasan, sehingga dapat melihat perbedaan hasil analisa suatu kesatuan dalam citra objek.

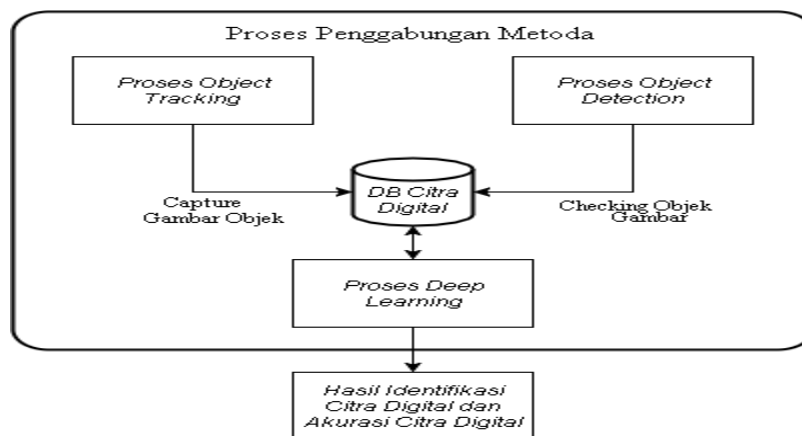
2. Metode Penelitian



Gambar 1 Kerangka Berpikir

Kegiatan penelitian ini terdiri dari beberapa tahap yaitu tahap penelitian, dan tahap perancangan dan implementasi seperti dapat dilihat pada Gambar 1 kerangka berpikir. Pada tahap penelitian dimulai dengan menentukan perumusan masalah dan identifikasi tujuan. Studi literatur dilakukan untuk memperdalam pemahaman mengenai cara kerja dari masing- masing metode yakni *Deep Learning*, *CAMShift*, dan *Faster R-CNN* dengan mengumpulkan referensi-referensi berupa buku dan jurnal terkait dengan penelitian.

Selanjutnya pada tahap perancangan dan implementasi dilakukan pembagian tugas terhadap semua proses yang ada seperti yang dapat terlihat pada Gambar 2 proses penggabungan metoda. Mulai dari objek *detection*, *object tracking*, *deep learning* hingga keluaran hasil. Setelah itu dilakukan implementasi untuk setiap proses yang digunakan dengan *Raspberry Pi* dan bantuan *Python* dan *Library OpenCV* untuk menghitung serta menjalankan setiap proses yang ada. Pada tahap proses penggabungan metoda untuk masing- masing metode, dilakukan kegiatan mengkombinasikan *Faster R-CNN* dengan *CAMShift*. Selanjutnya proses penggabungan metoda dibangun menggunakan data *real time video* yang telah dikumpulkan. Untuk lebih jelasnya tahapan penggabungan metoda bisa dilihat dari bagan dibawah ini.



Gambar 2 Proses Penggabungan Metoda

Proses penggabungan metoda terdiri dari beberapa tahap:

1. Pada proses *Tracking* dilakukan pelacakan objek citra digital yang akan diidentifikasi setelah itu melakukan *capture* terhadap citra digital tersebut dan data yang di *capture* tersebut disimpan dalam file database. (bisa dilihat Gambar 5 untuk lebih jelas)
2. Pada proses *Detection* akan mendeteksi objek citra digital yang telah di *capture* sebelumnya apakah data tersebut ada didalam database atau tidak. Jika data tersebut ada maka dilanjutkan dengan proses *deep learning*. Jika tidak maka tak akan melakukan eksekusi apa-apa. (bisa dilihat Gambar 6 untuk lebih jelas)
3. Proses *Deep Learning* ini melakukan tugasnya yakni melakukan kegiatan perhitungan algoritma *CAMShift* dan *Faster R-CNN* dari data citra digital yang ada didalam database positif. (bisa dilihat Gambar 7 untuk lebih jelas)
4. Hasil analisa tersebut didapatkan dari pemanfaatan penggabungan *Deep Learning*, *CAMShift* dan *Faster R-CNN*. (bisa dilihat Gambar 8 lebih jelas)

Selanjutnya menggunakan bantuan arsitektur *deep learning* terlihat pada gambar 3 arsitektur *deep learning* yakni dengan dilakukan evaluasi kinerja dari segi akurasi terhadap masing-masing proses yang telah dilakukan menggunakan data video yang telah dikumpulkan. Pada tahap terakhir yaitu menarik kesimpulan atas pengujian dan pembahasan yang telah dilakukan dan memberikan saran berdasarkan percobaan dan pengujian yang telah dilakukan terlihat pada gambar 7 dan 8.

### 3. Hasil dan Pembahasan

#### a. Arsitektur *Deep Learning*

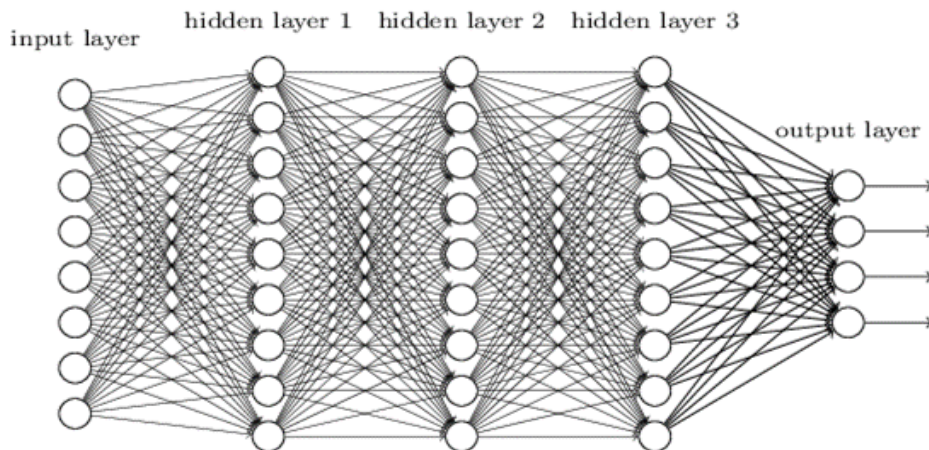
Dilihat dari gambar 3 arsitektur *deep learning* bisa dijelaskan tahapan proses yang nantinya akan diolah sedemikian rupa dengan bantuan *deep learning* yang terdiri dari :

##### 1. *Data Storage*

*Data storage* merupakan penyimpanan seluruh pelatihan citra digital yang nantinya akan di olah *layer*. [6],[8]

##### 2. *Layer*

*Layer* disini bertugas sebagai tempat proses pengolahan citra digital yang telah ada didalam data storage untuk dibandingkan sehingga mendapatkan kesimpulan apa yang terdapat dalam citra digital tersebut. [9],[12]



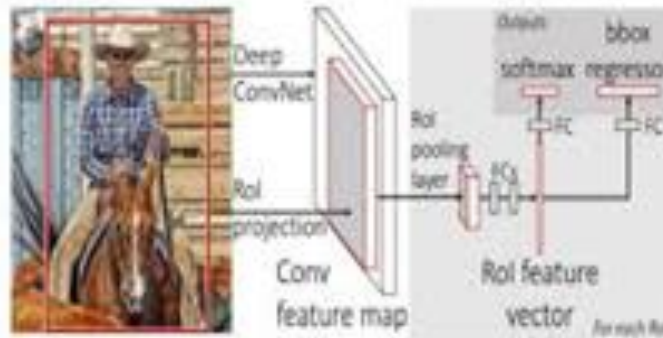
Gambar 3 Arsitektur *Deep Learning*

#### b. Arsitektur *CAMShift*

*CAMShift* (*Continuously Adaptive Mean-shift*) adalah algoritma pelacakan pergerakan yang merupakan perluasan dari algoritma *Mean-shift* yang diusulkan oleh Bradski & Clara pada tahun 1998. Merupakan teknik *non-parametric* yang mendaki suatu kepadatan pada distribusi probabilitas untuk menemukan puncak dari suatu distribusi. Ide utamanya adalah dengan melakukan operasi *mean-shift* terhadap semua *frame* dalam urutan video, menggunakan pusat massa dan ukuran dari *search window* yang diperoleh dari *frame* sebelumnya sebagai nilai inisial untuk *search window* pada *frame* selanjutnya, dan mencapai pelacakan objek dengan proses secara iterasi. [2],[4],[5]

c. **Arsitektur *Faster R-CNN***

*Faster R-CNN* merupakan pengembangan algoritma *Fast R-CNN* yang menghasilkan kecepatan dalam pelatihan data dan kecepatan dalam pengenalan objek. [11]



Gambar 4 Proses *Faster R-CNN*

d. **Pembahasan**

a. **Tahapan objek *tracking***

Sebelum mulai melatih data training positif hal yang dilakukan adalah menghitung intensitas piksel rata-rata di semua gambar atau video dalam rangkaian untuk masing-masing saluran Merah, Hijau, dan Biru atau biasa di kenal dengan RGB (*Red, Green* dan *Blue*). Sehingga bisa dirumuskan dengan

$$“\mu R, \mu G, \text{ dan } \mu B”$$

Atau bisa juga dalam beberapa kasus, nilai merah, hijau, dan biru rata-rata dapat dihitung secara bijaksana dari pada pixel-wise, menghasilkan matriks

$$“[M] \times [N]”$$

Dalam hal ini matriks  $[M] \times [N]$  untuk setiap saluran matriks akan dikurangkan dari gambar atau video masukan selama pelatihan ataupun pengujian.

Kedua metode tersebut merupakan bentuk pengurangan rata-rata yang benar-benar valid. Namun, cenderung melihat versi *pixel-wise* yang digunakan lebih sering, terutama untuk *dataset* yang lebih besar. Terlihat pada gambar 5 *Objek Tracking* ditandai dengan garis merah yang berfungsi untuk menandakan pergerakan suatu objek pada citra digital.



Gambar 5 *Objek Tracking*

**b. Tahapan objek *detection***

Setelah melakukan training data hal berikutnya adalah siap untuk melewati sebuah tahapan pengujian dimana *RGB pixel* gambar atau video tersebut diuji dengan metode normalisasi dan skala *vector*. Sehingga bisa dirumuskan untuk normalisasi seperti berikut :

$$\begin{aligned} & \text{“R=R-}\mu\text{R, G=G-}\mu\text{G, B=B-}\mu\text{B”} \\ & \text{dan untuk skala vector sebagai berikut} \\ & \text{“R=(R-}\mu\text{R)/}\sigma\text{, G=(G-}\mu\text{G)/}\sigma\text{, B=(B-}\mu\text{B)/}\sigma\text{”} \\ & \text{“dimana } \sigma = 1 \text{”} \end{aligned}$$

Dapat dilihat dari gambar 6 proses objek *detection* yang mana bulatan kuning tersebut menandakan objek yang dikenali. Semakin dekat objek tersebut maka semakin baik dan bagus identifikasinya.

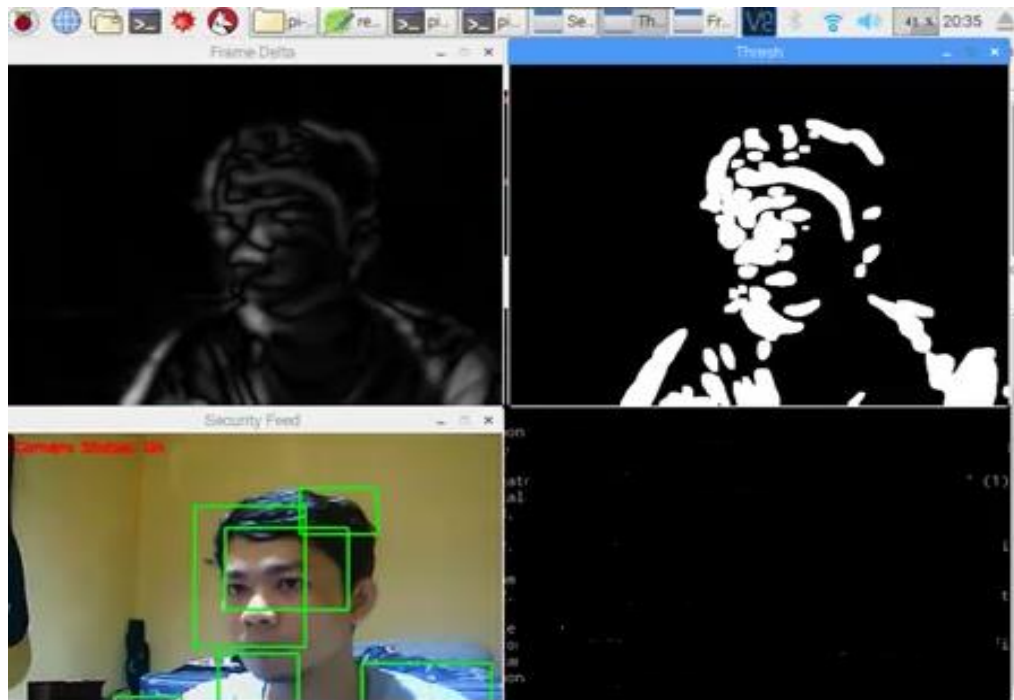


Gambar 6 Proses Objek Detection

**c. Tahapan *Deep Learning***

Setelah semua data tersimpan maka tugas *deep learning* disini adalah menyimpan hasil *capture*, mengolah hasil *capture* dan mengevaluasi hasil *capture* gambar. Tahapan itu sendiri dimulai dengan

- Input data (Proses *Tracking*)
- Training data (Proses *Detection*)
- Proses data (Proses *Deep Learning*)
- Hasil proses data (Penggabungan Metoda)



Gambar 7 Proses Pengolahan citra dengan Deep Learning



Gambar 8 Hasil Proses Pengolahan

#### 4. Simpulan

Berdasarkan hasil studi yang dilakukan maka dapat disimpulkan untuk menggabungkan metode tersebut sangatlah bisa, akan tetapi masih jauh dari kata sempurna. Kekurangan dari hasil studi ini adalah pada saat proses penyimpanan data digital sangat banyak maka akan memakan proses yang lama. Untuk akurasi citra gambar digital sendiri masih belum mencapai kata memuaskan. Besar harapan kedepannya dapat dilakukan penyempurnaan dari kegiatan studi penelitian ini.

#### Daftar Pustaka

- [1] W. Jiang-tao and Y. Jing-yu, "Object tracking based on kalman-mean shift in occlusions," *Journal of System Simulation*, pp. 4216–4220, 2007.

- [2] J. Xia, J. Wu, H. Zhai, and Z. Cui, “Moving vehicle tracking based on double difference and CAMShift,” Proceedings of the 2009 International Symposium on Information Processing, pp. 29–32, 2009.
- [3] M. K. Chouhan, R. Mishra, and D. D. Nitnawwre, “Movable object tracking by using mean shift method with adjusted background histogram,” International Journal of Advanced Research in Computer Science and Software Engineering, pp. 16–19, 2012.
- [4] Li, Dawei, Lihong Xu, and Yang Wu. (2017). *Improved CAMShift object tracking based on Epanechnikov Kernel Density Estimation and Kalman filter*.
- [5] Tian, Yun, Carol Taylor, and Yanqing Ji. (2018). *Improving the Performance of the CamShift Algorithm Using Dynamic Parallelism on GPU*.
- [6] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell. *Caffe: Convolutional Architecture for Fast Feature Embedding\**. 2014.
- [7] D. Ciresan, U. Meier, J. Masci, L. Gambardella, and J. Schmidhuber. *High-Performance Neural Networks for Visual Object Classification*. In: CoRR abs/1102.0183 (2011).
- [8] D. Parikh and C.L. Zitnick. *Finding the weakest link in person detectors*. In Computer Vision and Pattern Recognition, pages 1425–1432. IEEE, 2011.
- [9] T. Malisiewicz, A. Gupta, and A.A. Efros. *Ensemble of exemplar-svms for object detection and beyond*. In International Conference on Computer Vision, pages 89–96. IEEE, 2011.
- [10] Zeiler, Matthew D. and Fergus, Rob. Stochastic pooling for regularization of deep convolutional neural networks. In *International Conference on Learning Representations*, 2013.
- [11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. CVPR, 2014.
- [12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. *Decaf: A deep convolutional activation feature for generic visual recognition*. ICML, 2014.