

PELACAKAN OBJEK BERBASIS PADA MODEL MULTI-DOMAIN CONVOLUTIONAL NEURAL NETWORK DENGAN MENGGUNAKAN METODE DECISION TREE

Kautsar Fadly Firdaus¹, Suryo Adhi Wibowo², Nur Andini³)

Fakultas Teknik Elektro Universitas Telkom
Jl. Telekomunikasi No.1, Terusan Buah Batu, Jawa Barat, Indonesia
Email: kautsarfadlyfirdaus@gmail.com¹), suryoadhiwibowo@telkomuniversity.ac.id²),
nurandini@telkomuniversity.ac.id³)

Abstrak

Object Tracking menjadi salah satu masalah utama di *computer vision*. Pelacakan objek digunakan untuk memudahkan dan mempercepat proses pengolahan data, mulai dari pendeteksian objek dan *human-computer* hingga pencitraan medis. Pada penelitian ini dilakukan pengujian algoritma yang dikembangkan dari metode *Multi-Domain Convolutional Neural Network (MDNet)*. *MDNet* merupakan sebuah metode yang menggabungkan dua domain, yaitu domain-spesifik dan domain-independen. Pada domain-independen terdapat *shared layer* yang menyimpan nilai informasi generik seperti *illumination variation*, *scale variation*, *blur* dan *background clutter*. Sedangkan pada domain-spesifik adalah cabang – cabang domain sebanyak *K* (banyak data) yang terpisah satu sama lain, pada domain ini terjadi *binary classification* untuk menentukan target pada setiap domain. Pelacakan target akan dilakukan dengan memberikan kandidat *bounding box* acak. Ekstraksi fitur akan dilakukan dengan mengambil ciri pada domain-independen dan domain-spesifik yang kemudian akan diklasifikasi oleh *Decision Tree* berdasarkan aturan – aturan yang telah ditetapkan sebelumnya. Performa algoritma pelacakan objek diukur berdasarkan *precision plot* dan *success plot* pada *One Pass Evaluation (OPE)*. Algoritma yang diajukan pada penelitian ini menunjukkan performa yang lebih baik dengan peningkatan sebesar 0.087 poin pada *precision plot* dan 0.071 poin pada *success plot* dibandingkan *state-of-the-art* pelacakan objek yang telah ada.

Kata kunci: *Object Tracking*, *CNN*, *MDNet*, *Decision Tree*.

1. Pendahuluan

Perkembangan teknologi yang sangat pesat di era globalisasi menciptakan tantangan tersendiri, berbagai teknologi dirancang untuk memudahkan kebutuhan manusia, salah satunya adalah pelacakan objek. Pelacakan objek merupakan salah satu sub materi yang penting di bidang *computer vision* dalam berbagai aplikasi mulai dari pendeteksian objek dan *human-computer* hingga pencitraan medis. Meskipun pelacakan

objek telah diteliti cukup lama dan telah mengalami peningkatan yang cukup pesat, pelacakan objek masih memiliki beberapa masalah. Banyak faktor memengaruhi performa dari algoritma pelacakan objek, termasuk *illumination variation*, *scale variation*, *occlusion*, *deformation*, *motion blur*, *fast motion*, *in-plane rotation*, *out-of-plane rotation*, *out-of-view*, *background clutter*, dan *low resolution* yang sampai saat ini belum ditemukan satu solusi untuk menyelesaikan masalah pada semua skenario [1]. Algoritma pelacakan objek dikelompokkan menjadi dua bagian, yaitu *generative* dan *discriminative*. *Generative* adalah sebuah metode yang menggunakan *generative models* lalu mencocokkan area target dengan model terbaik yang sesuai, sedangkan *discriminative* adalah sebuah metode yang membedakan antara objek dan latar [2].

Pada penelitian sebelumnya pelacakan objek telah dilakukan dengan berbagai metode. Pada *Tracking Learning Detection* dirancang suatu sistem dengan tiga komponen dasar, yaitu *tracking*, *learning*, dan *detection* yang dapat melakukan pembaruan nilai komponen pada setiap pemrosesan citra oleh *P-N Learning*, tetapi memiliki penurunan performansi ketika nilai *eigen* semakin besar [3]. Pada *Robust Object Tracking with Online Multiple Instance Learning* yang menghasilkan tingkat *robustness* cukup baik, tetapi jika objek keluar dari frame dan masuk kembali maka objek tidak dapat dikenali lagi [4]. Sedangkan pada *Hierarchical Convolutional Features for Visual Tracking* memanfaatkan ekstraksi ciri pada setiap layer namun mengalami kegagalan ketika tracklet tidak berhasil mengikuti objek dan tidak efektif ketika menggunakan ciri semantik untuk membedakan antara bagian depan gelap dengan latar terang [5].

Termotivasi oleh fakta maka digunakan sebuah algoritma pelacakan objek menggunakan *pretrained Multi-Domain Convolutional Neural Network* dengan pengklasifikasian berbasis *Decision Tree*. *Convolutional Neural Network (CNN)* adalah sebuah metode dalam *deep learning* yang digunakan untuk klasifikasi citra untuk segmentasi semantik, deteksi objek, dan ekstraksi fitur [6], [7]. *CNN* dikenal memiliki performansi yang sangat baik dalam mengolah citra. *CNN* terdiri dari beberapa layer yang skemanya dapat disusun sesuai dengan kebutuhan pengguna. Layer – layer yang

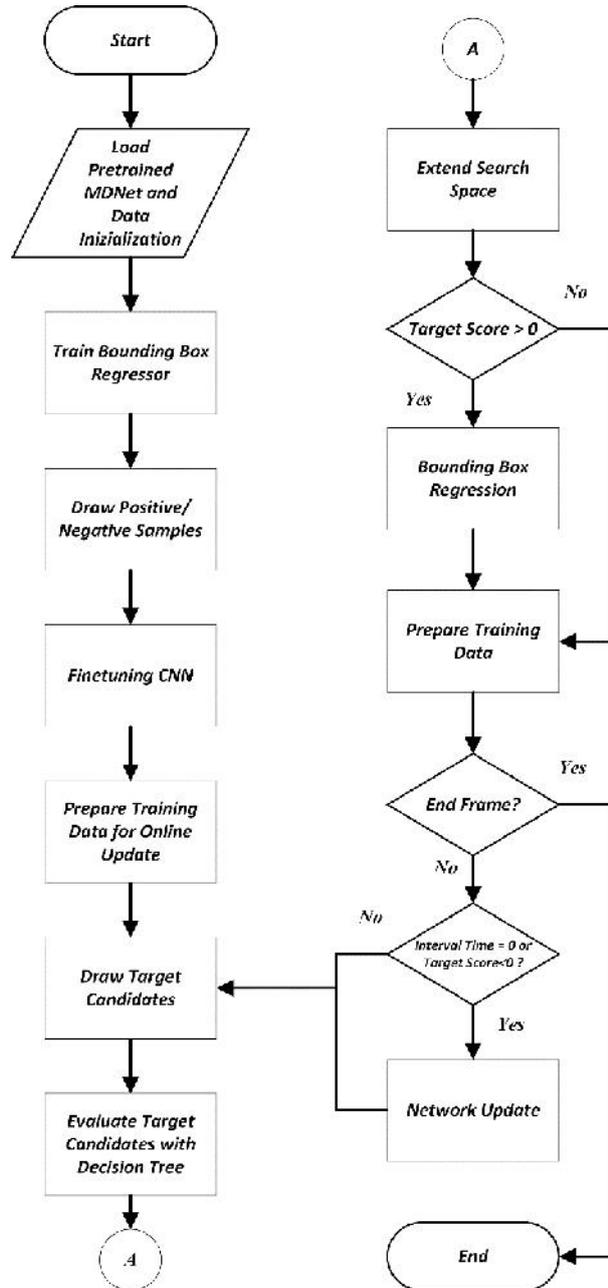
dimaksud adalah *convolutional*, *ReLU*, *pooling*, dan *fully connected* yang memiliki fungsi masing – masing. Susunan layer dari CNN dapat mempengaruhi performa. Sedangkan *Decision Tree* adalah sebuah non-parametrik *supervised machine learning* yang digunakan untuk pengklasifikasian dan operasi regresi. Tujuan dari *Decision Tree* adalah membuat prediksi dari input berdasarkan aturan – aturan dari masukan sebelumnya [8], [9]. *Decision tree* memiliki performa yang baik dan waktu komputasi yang cukup singkat. Penelitian ini menggunakan MDNet untuk ekstraksi ciri dimana domain-spesifik dan domain-independen akan dipisahkan. Domain-independen bertugas untuk mengambil informasi generik *robustness* pada citra, seperti *illumination variation*, *scale variation*, *occlusion*, *deformation*, *motion blur*, *fast motion*, *in-plane rotation*, *out-of-plane rotation*, *out-of-view*, *background clutter*, dan *low resolution*. Sedangkan pada domain-spesifik akan terjadi *binary classification* yang berfungsi untuk membedakan antara objek dan latar pada citra karena setiap domain memiliki nilai objek dan latar yang berbeda [2]. Rancangan sistem MDNet memiliki masukan 107 x 107 citra RGB, dan memiliki lima *hidden layers* termasuk tiga *convolutional layers* (conv 1-3) dan dua *fully connected layers* (fc4-5). Sistem juga memiliki K buah cabang untuk *fully connected layer* (fc6¹-fc6^k) yang merepresentasikan K domain sebagai *training sequences*. Kedua *fully connected layer* (fc4-5) memiliki keluaran 512 unit yang dikombinasikan dengan *ReLUs* dan *dropouts*. Setiap cabang K memiliki *binary classification layer* dengan *softmax cross-entropy loss* yang berfungsi untuk membedakan objek dan latar di setiap region. Layer fc6¹-fc6^k merupakan domain-spesifik dan layer – layer sebelumnya merupakan *shared layers*. Nilai – nilai yang telah didapatkan dari ekstraksi ciri akan diklasifikasikan oleh *Decision Tree classifier*.

2. Pembahasan

Desain Model Sistem

Pada penelitian ini dibuat sebuah pemodelan sistem deteksi objek bergerak dengan data berupa sekuen citra. Skema dari penelitian ini menggunakan arsitektur MDNet dengan klasifikasi *Decision Tree* pada evaluasi kandidat target dan lokasi. *Decision Tree* digunakan untuk melakukan klasifikasi data masukan dengan aturan – aturan yang telah ditetapkan berdasarkan data latih masukan – masukan sebelumnya untuk pemilihan lokasi *bounding box* yang optimal. Nilai ciri yang dimaksud berupa lokasi *bounding box* dan *target score* yang mendekati ciri sebelumnya. Arsitektur desain sistem secara umum dapat dibagi menjadi dua bagian utama, yaitu inialisasi dan iterasi. Pada inialisasi terdapat proses *load pretrained MDNet and data initialization*, *train bounding box regressor*, *draw positive/negative samples*, *finetuning CNN*, *prepare training data for online update*. Sedangkan pada proses iterasi terdapat *draw target candidates*, *evaluates target candidates with Decision Tree*, *extend search space*, *bounding box regression*, *prepare training data*, dan *network update*.

Proses iterasi akan dijalankan mulai dari frame ke-2 hingga frame terakhir dari sekuen data. Adapun desain model umum yang akan dibuat sebagai berikut :



Gambar 1. Diagram Alir sistem EC-CNN.

a. Load Pretrained MDNet and Data Inizialization

Data *pre-trained MDNet*, informasi citra (nama, ukuran, intensitas, dll.), jumlah citra dalam satu sekuen, dan *ground truth bounding box* akan diinisialisasi untuk diteruskan ke proses selanjutnya. Pada proses ini juga akan membaca informasi mengenai domain-spesifik dan domain-independen. Proses ini akan dilakukan satu kali, yaitu pada saat dimulainya sistem.

b. Train Bounding Box Regressor

Sampel *bounding box* acak akan dibuat dari data yang telah diinisialisasi. 10000 sampel *bounding box* akan dibuat dengan masukan *ground truth* awal, setelah itu

nilai sampel yang memiliki irisan $r > 0.6$ akan disimpan dan akan dipilih 1000 *sample* secara acak. Pada proses ini akan dilakukan ekstraksi ciri yang menghasilkan matriks 4-Dimensi, setelah itu matriks *ground truth* akan direplikasi sebanyak 1000 kali untuk melatih data *bounding box regressor*.

c. Draw Positive / Negative Samples

Pada proses *positive sampling* mirip seperti proses sebelumnya, yaitu akan dibangkitkan 1000 *bounding box* dengan masukkan *ground truth* awal, setelah itu nilai sampel disimpan dan akan dipilih 500 sampel *bounding box* yang memenuhi irisan $r > 0.7$ secara acak. Sedangkan pada proses *negative sampling* akan dibangkitkan 1000 sampel *bounding box* dengan masukan *ground truth* awal, setelah itu nilai sampel disimpan dan akan dipilih 5000 sampel *bounding box* yang memenuhi irisan $r < 0.5$ secara acak. *Positive samples* dan *negative samples* akan membentuk matriks baru, kemudian nilai fitur dari masing – masing sampel akan diambil sesuai dengan jumlahnya, yaitu 500 *positive samples* dan 5000 *negative samples*.

d. Finetuning CNN

Fully Connected Layer (fc_6) akan terbentuk pada proses ini dimana nilai tersebut akan digunakan pada *binary classification* untuk memisahkan antara latar dan objek. Nilai *shared layer* akan diperbarui, dimana *shared layer* merepresentasikan *robustness* pada sekuen data.

e. Prepare Training Data for Online Update

Pada proses ini akan dilakukan inisiasi data positif dan negatif sejumlah citra pada sekuen. *Negative samples* dibangkitkan dengan masukkan *ground truth* awal, setelah itu nilai *bounding box* disimpan dan akan dipilih 200 *bounding box* yang memenuhi $r < 0.5$ secara acak. *Negative samples* dan *positive samples* akan disatukan menjadi matriks baru yang kemudian akan dilakukan ekstraksi ciri pada proses ini.

f. Draw Target Candidates

Bounding box akan dibangkitkan dengan jumlah 256 sampel. Setelah sampel – sampel dibangkitkan, ekstraksi ciri dilakukan dengan masukan sampel – sampel sebelumnya dan citra iterasi ke - n untuk menghasilkan matriks ciri 4-D.

g. Evaluate Target Candidates with Decision Tree

Nilai matriks ciri yang didapat selanjutnya akan diambil *target score* yang berjumlah 256, semakin tinggi *target score* maka semakin mirip *bounding box* dengan nilai *bounding box* sebelumnya. Sampel – sampel yang telah dibangkitkan dan *target score* akan menjadi nilai masukan *Decision Tree classifier* untuk membentuk aturan – aturan baru di setiap iterasi. Setelah itu prediksi akan dilakukan dengan masukkan *Decision Tree* yang telah dibangkitkan pada iterasi ke-n dan *bounding box* sebelumnya menghasilkan nilai *bounding box* baru. Pada proses ini terdapat dua buah kandidat *bounding box*, *bounding box* yang memiliki jarak terdekat ke *bounding box* sebelumnya akan dipilih sebagai *bounding box* yang baru dan menjadi hasil akhir sementara. Adapun

pemilihan jarak terdekat *bounding box* menggunakan rumus:

$$\text{jarak} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

Dimana x merupakan koordinat horizontal dan y merupakan koordinat vertikal.

h. Extend Search Space

Jika *target score* < 0 maka nilai konstanta translasi *sampling* dirubah menjadi $\text{trans_f} = \min(1.5, 1.1 * \text{trans_f})$ dimana $\text{trans_f} = 0.6$, tetapi jika *target score* > 0 maka $\text{trans_f} = 0.6$.

i. Bounding Box Regression

Jika *target score* > 0 maka akan dilakukan regresi *bounding box*. Sampel *bounding box* dari nilai lima fitur teratas dari *bounding box* sebelumnya yang memiliki nilai tertinggi akan dibangkitkan, setelah itu akan dilakukan regresi *bounding box* untuk mendapatkan lima buah *bounding box* baru. Hasil akhir pada proses ini adalah rata – rata dari nilai regresi *bounding box*.

j. Prepare Training Data

Jika *target score* > 0 maka akan dilakukan pengambilan *positive samples* dan *negative samples*. *Positive samples* dibangkitkan seperti proses *training data* awal yang berjumlah lima puluh sampel *bounding box*. *Positive samples* baru akan dipilih secara acak sejumlah nilai *positive samples* sebelumnya yang memiliki irisan $r > 0.7$ dari *bounding box* sebelumnya. Sedangkan *negative samples* memiliki proses seperti pada *training data*, tetapi nilai *negative samples* yang memiliki irisan $r < 0.2$ sebanyak 200 sampel. Nilai ekstraksi ciri akan diambil berdasarkan citra dan sampel yang telah dibangkitkan sebelumnya

k. Network Update

Dalam pelacakan objek, pembaruan model perlu dilakukan karena terjadinya perubahan – perubahan pada target. Hal ini dilakukan dengan dua cara, yaitu *long-term update* dan *short-term update*. *Long-Term Update* akan melakukan *model update* pada interval frame yang telah ditentukan ($Tl=100$), *long-Term Update* digunakan untuk *robustness*. Sedangkan *short-term update* akan dilakukan ketika terjadi perubahan objek secara tiba – tiba, nilai *abrupt appearance changes* ($F+(x^*) < 0.5$), *short-term update* dilakukan untuk *adaptiveness*.

A. OTB-50 Benchmark Dataset

OTB-50 Benchmark Dataset adalah sebuah dataset pilihan dari *OTB-100* yang memiliki tingkat kerumitan yang tinggi dan biasa digunakan untuk *in-depth analysis*. Pengujian performansi pada algoritma pelacakan objek sulit dilakukan karena banyak faktor yang memengaruhi hasil pengujian. Untuk melakukan analisis kelebihan dan kekurangan pada algoritma pelacakan objek, maka *benchmark dataset* akan dibagi berdasarkan 11 atribut. Setiap atribut merepresentasikan permasalahan spesifik yang menjadi faktor penting pada pelacakan objek. Bisa saja satu data memiliki banyak atribut dan bisa saja satu atribut lebih sering muncul dibandingkan atribut lain [1].

Atribut yang dimaksud adalah *illumination variation*, *scale variation*, *occlusion*, *deformation*, *motion blur*, *fast motion*, *in-plane-rotation*, *out-of-plane*, *out-of-view*, *background clutter*, dan *low resolution* yang dapat dilihat pada tabel 1.

Tabel 1. Penjelasan Data Atribut [1].

Atribut	Deskripsi
IV	<i>Illumination Variation</i> merupakan pencahayaan pada suatu area objek yang berubah secara signifikan.
SV	<i>Scale Variation</i> merupakan rasio keluarnya nilai di atas batas pada bounding box frame pertama dan frame yang berjalan. $[1/t_s, t_s], t_s > 1 (t_s=2)$
OCC	<i>Occlusion</i> merupakan keberadaan objek yang terhalangi sebagian atau seluruhnya.
DEF	<i>Deformation</i> – perubahan citra secara <i>non-rigid deformation</i> .
MB	<i>Motion Blur</i> merupakan pengkaburan citra pada suatu area akibat pergerakan objek atau kamera.
FM	<i>Fast Motion</i> – Pergerakan dari <i>ground – truth</i> lebih besar dari t_m piksel ($t_m=20$).
IPR	<i>In-Plane Rotation</i> – Rotasi objek ke dalam pada <i>image plane</i> .
OPR	<i>Out – of – Plane Rotation</i> objek ke luar pada <i>image plane</i> .
OV	<i>Out-of-View</i> merupakan keluarnya sebagian dari objek pada <i>frame</i> citra
BC	<i>Background Clutters</i> merupakan kejadian dimana objek memiliki tekstur atau warna yang mirip dengan latar
LR	<i>Low Resolution</i> – Jumlah piksel pada <i>ground – truth bounding box</i> kurang dari t_r ($t_r=400$)

B. Parameter Performansi

Parameter performansi merupakan standar pengujian algoritma pelacakan objek dengan parameter – parameter yang telah ditentukan. Adapun parameter – parameter yang digunakan adalah *precision plot* dan *success plot* pada metode *One Pass Evaluation* (OPE) dan *Area Under Curve* (AUC).

a. Precision Plot

Precision Plot digunakan untuk melakukan pengukuran jarak rata - rata *euclidean* antara titik tengah lokasi pelacakan objek oleh sistem dan titik tengah *ground truth* yang diletakan secara manual yang dilakukan di semua frame. Ketika algoritma pelacakan objek kehilangan objek target, maka lokasi target dapat menjadi acak, oleh karena itu nilai rata – rata *error* tidak dapat mengukur performansi pelacakan objek dengan benar. Berdasarkan alasan tersebut frame yang memiliki nilai lokasi *ground truth* yang berada dibawah nilai batas akan digunakan untuk mengukur performansi pelacakan objek.

Nilai titik tengah lokasi *error* pada *precision plot* hanya dapat mengukur perbedaan jumlah piksel (jarak piksel) dan tidak merepresentasikan ukuran dan skala dari objek target.

b. Success Plot

Success Plot merupakan salah satu pengujian yang menggunakan nilai irisan [10]. Pada lokasi *bounding box* objek r_t dan lokasi *ground truth* r_0 bertumpuk maka nilai irisan dapat didefinisikan:

$$S = \frac{|r_t \cap r_0|}{|r_t \cup r_0|} \tag{2}$$

Dimana \cap merepresentasikan irisan, \cup merepresentasikan gabungan, dan $|:|$ merepresentasikan jumlah piksel. Nilai rata – rata dari irisan (AOS) dapat dijadikan sebagai pengukuran performansi. Nilai irisan juga dapat menentukan apakah suatu algoritma berhasil melakukan pelacakan objek pada satu frame, dengan melakukan pengecekan nilai S lebih besar dari nilai batas t_0 .

c. One Pass Evaluation

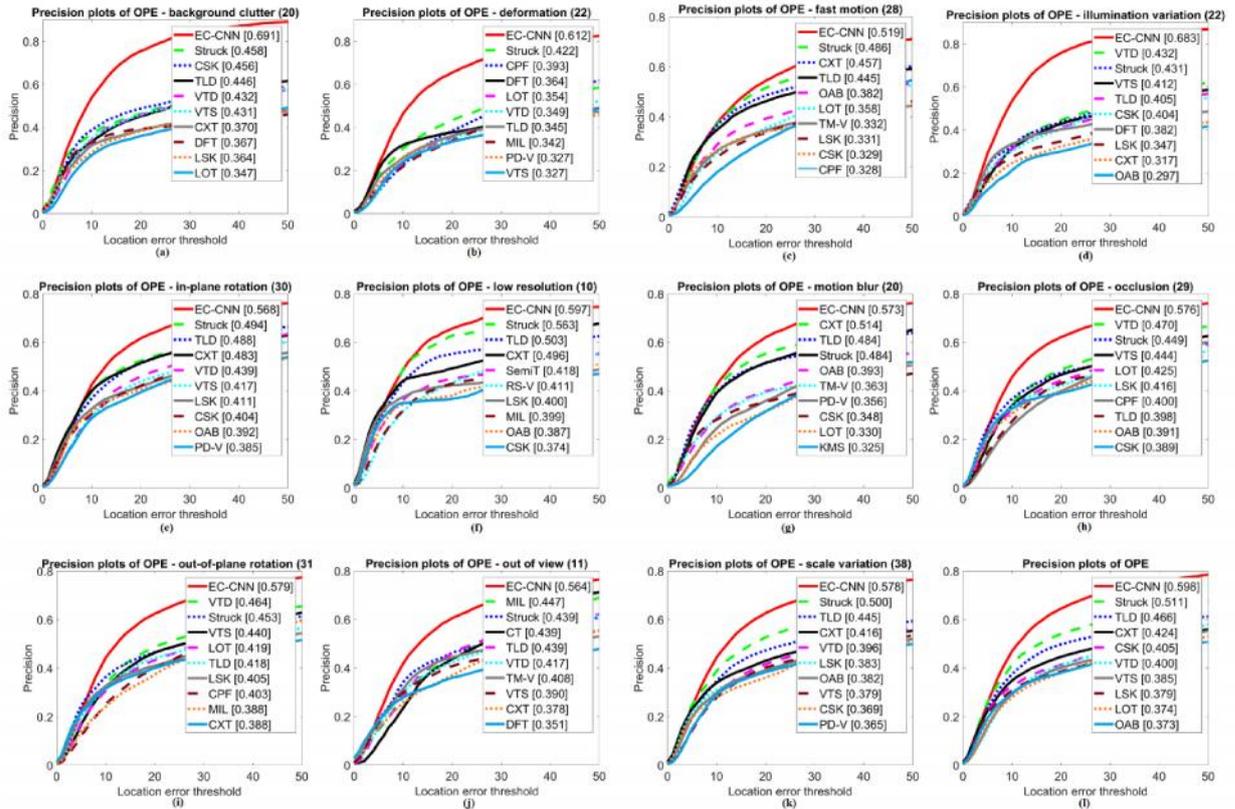
One Pass Evaluation (OPE) merupakan sebuah metode pengujian dimana ketika memulai algoritma pelacakan dengan meletakkan lokasi *ground truth* objek pada frame pertama dan melakukan pengukuran nilai rata – rata dari *precision plot* atau *success plot* pada semua frame.

d. Area Under Curve

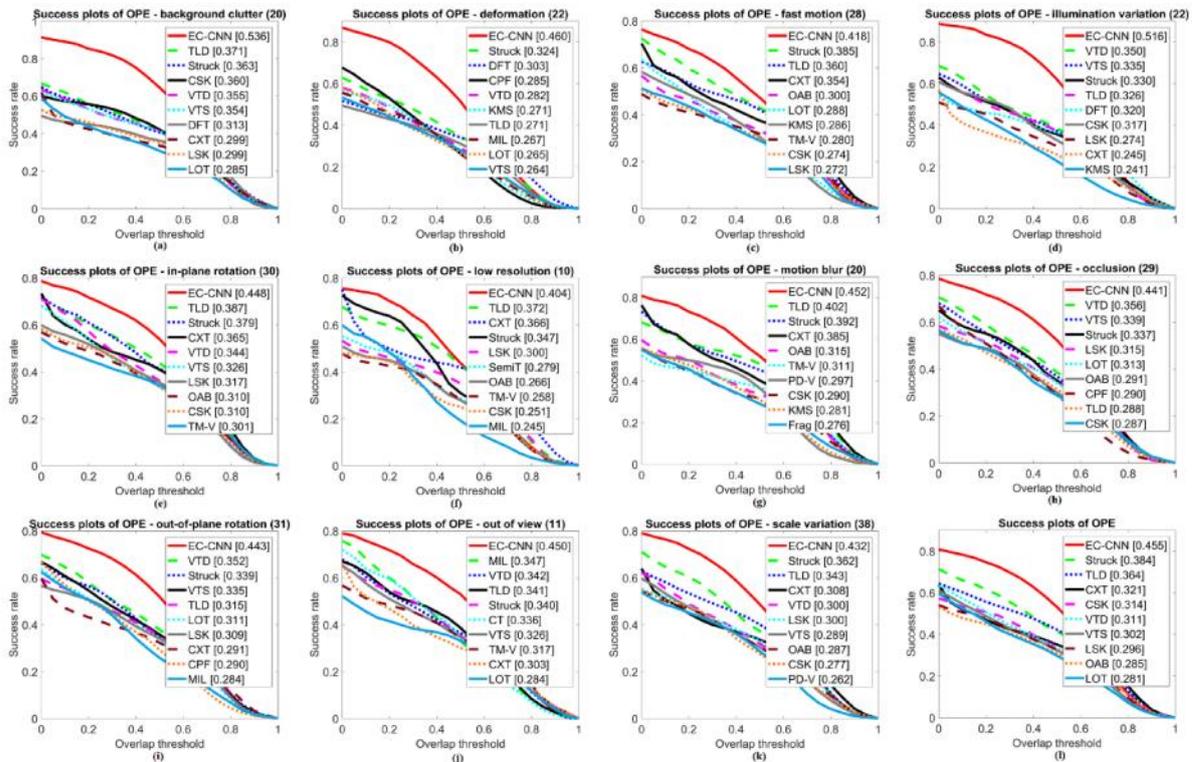
Area Under Curve (AUC) merupakan sebuah metode pengujian dimana nilai rata – rata dari *success plot* yang melebihi nilai batas. Nilai AUC pada satu data sebanding dengan nilai AOS pada dataset. Dengan kata lain nilai rata – rata dari frame rasio yang nilai irisannya melebihi nilai batas sebanding dengan nilai rata – rata dari irisan.

C. Hasil Pengujian

Pada bagian ini dilakukan pengujian secara komprehensif algoritma EC-CNN dengan *OTB-50 Benchmark Dataset* menggunakan *precision plot* dan *success plot* pada *one pass evaluation* dan *area under curve* dengan 29 *state-of-the-art object tracking*. Algoritma pelacakan objek yang dimaksud adalah [1] : VR, TM, RS, PD, MS, CT, TLD, IVT, DFT, ASLA, LIAPG, ORIA, MTT, CSK, SCM, LOT, CPF, Struct, MIL, OAB, SBT, BSBT, FRAG, KMS, SMS, LSK, VTS, VTD, dan CXT. Pada gambar 2 terdapat grafik *precision plot* dan nilai rata – rata dari 11 atribut. Sedangkan pada gambar 3 terdapat grafik *success plot* dan nilai rata – rata dari 11 atribut. Pada grafik dapat dilihat bahwa 10 dari 30 algoritma pelacakan objek terbaik ditampilkan pada grafik, EC-CNN menunjukkan peningkatan performa yang lebih baik dengan peningkatan sebesar 0.087 poin pada *precision plot* dan 0.072 poin pada *success plot* dari *state-of-the-art object tracking* yang telah disebutkan di atas.



Gambar 2. Precision plot dari 11 atribut: background clutter, deformation, fast motion, illumination variation in-plane rotation, low resolution, motion blur, occlusion, out-of-plane rotation, out of view, dan scale variation serta quality of precision plot.



Gambar 3. Success plot dari 11 atribut: background clutter, deformation, fast motion, illumination variation in-plane rotation, low resolution, motion blur, occlusion, out-of-plane rotation, out of view, dan scale variation serta quality of success plot.

3. Kesimpulan

Pada paper ini diajukan algoritma pelacakan objek berbasis CNN yang dilatih pada jaringan multi-domain dengan klasifikasi kandidat *bounding box* menggunakan *Decision Tree*. EC-CNN memiliki dua domain, yaitu domain-independen dan domain-spesifik. Pada domain-independen terdapat *shared layer* yang menyimpan informasi generik atau *robustness*, sedangkan pada domain-spesifik terjadi *binary classification*. EC-CNN menunjukkan performa yang lebih baik dengan peningkatan sebesar 0.087 poin pada *precision plot* dan 0.071 poin pada *success plot* dibandingkan dengan *state-of-the-art object tracking* yang telah ada pada *OTB-50 Benchmark Dataset*.

Daftar Pustaka

- [1] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [2] H. Nam and B. Han, "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking," *arXiv Prepr. arXiv1510.07945*, pp. 4293–4302, 2015.
- [3] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [4] B. Babenko, Ming-Hsuan Yang, and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2011.
- [5] C. Ma, J. Bin Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 3074–3082, 2015.
- [6] J. Wu, "Introduction to Convolutional Neural Networks," pp. 1–28, 2016.
- [7] C.-C. J. Kuo, "Understanding Convolutional Neural Networks with A Mathematical Model," *Nips 2016*, no. 3, pp. 1–23, 2016.
- [8] B. Shepherd, "An Appraisal of a Decision Tree approach to Image Classification." * B. A. Shepherd. Machine Intelligence Research Unit, University of Edinburgh, U.K.," 2002.
- [9] V. Podgorelec and M. Zorman, "Decision Tree Learning," *Encyclopedia of Complexity and Systems Science*. pp. 1–28, 2015.
- [10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

Biodata Penulis

Kautsar Fadly Firdaus, Mahasiswa Teknik Telekomunikasi Universitas Telkom, angkatan 2014 dan peneliti di Laboratorium *Image Processing and Vision* (IMV) Universitas Telkom.

Suryo Adhi Wibowo, Dosen di Fakultas Teknik Elektro, Jurusan Teknik Telekomunikasi, Universitas Telkom Bandung dan peneliti di Laboratorium *Image Processing and Vision* (IMV) Universitas Telkom.

Nur Andini, Dosen di Fakultas Teknik Elektro, Jurusan Teknik Telekomunikasi, Universitas Telkom Bandung dan peneliti di Laboratorium *Image Processing and Vision* (IMV) Universitas Telkom.