PREDIKSI *Object Saliency* PADA VIDEO DENGAN PENDEKATAN *Convolutional Long Short Term Memory* DAN *Exponential Moving Average* DALAM ARSITEKTUR *Recurrent Neural Network*

A PREPRINT

Bernadetha Emma Wawin, Diana Petrina Santoso, Felicia Ferren
Statistics Department
BINUS University
Jakarta, Indonesia, 11530
(bernadetha.wawin, diana.santoso, felicia.ferren)@binus.ac.id

January 20, 2023

ABSTRACT

Salient object detection diadaptasi dari keterbatasan manusia untuk menangkap seluruh detail informasi penting dalam waktu singkat. Prediksi ini telah dikembangkan dalam berbagai metode. Namun, sebagian besar dari metode tersebut memiliki *temporal reccurences* yang kompleks. Oleh karena itu, penulis mengembangkan metode yang lebih sederhana yaitu dengan pendekatan *Exponential Moving Average* (EMA) pada *temporal state*. Hasil dari metode dengan tingkat kompleksitas yang lebih rendah ini, kemudian dibandingkan dengan metode yang memiliki tingkat kompleksitas tinggi, seperti *Convolutional Long Short Term Memory* (ConvLSTM). Setelah dilakukannya percobaan, ternyata hasil yang didapatkan oleh kedua metode ini berbeda signifikan dimana ConvLSTM gagal mendeteksi *object saliency* dengan nilai *loss training* mencapai 3.64. Sedangkan EMA mampu menampilkan bagian penting (saliensi) objek dan nilai loss training nya hanya 0.77. Karena perbedaan signifikan ini, serta tingkat kompleksitas EMA yang lebih rendah, maka metode EMA layak dipertimbangkan untuk dipelajari dan dikembangkan lebih lanjut untuk menyelesaikan *saliency task*.

Keywords Object Saliency · Temporal Reccurences · ConvLSTM · EMA

1 Introduction

Object Saliency adalah suatu bagian objek tertentu yang memiliki informasi lebih bermakna dan berkesan dibandingkan bagian-bagian lainnya [1]. Pada umumnya, kesan ini tercipta karena objek tersebut terlihat unik atau berbeda dari objek sekitarnya, misalkan berwarna kontras tinggi dan khas (*distinctive color*) [2]. Alhasil, bagian objek tersebut terlihat lebih menonjol dan menarik perhatian visual (*visual attention*) manusia.

Pada dasarnya, kemampuan dalam mendeteksi *object saliency* pada *computer vision* mengadaptasi dari keterbatasan manusia dalam memfokuskan daya persepsi dan kognitif ketika melihat suatu objek[3]. Keterbatasan visualisasi ini akan mengarahkan sebagian kecil informasi ke otak manusia dan terus berevolusi membentuk pola-pola memori di dalam otak manusia. Sebenarnya berkat keterbatasan ini, secara alamiah manusia akan lebih efisien dalam memproses dan menangkap informasi penting di tengah kehidupannya yang dikelilingi oleh objek visual yang sangat banyak. Oleh karena itu, sisi efisiensi inilah yang menjadi landasan perkembangan *computer vision* untuk mensimulasikan sistem visual *saliency* pada manusia secara artifisial. Harapannya, kemampuan ini dapat meningkatkan efisiensi sistem komputasi dalam berbagai bidang *computer vision* [4], salah satunya pada kasus *video saliency detection*. Kasus ini tergolong menarik karena sangat berguna untuk memahami perilaku yang mencuri perhatian manusia dan memiliki beberapa aplikasi praktis di dunia nyata seperti, *video captioning*, *compression*, *question answering*, *object segmentation*, *action recognition*[5], dan lain sebagainya. [6]

Saat ini, sudah banyak metode yang mengajukan pengembangan *salient object detection* (SOD) *task*. Salah satunya adalah dengan pendekatan *Convolutional LSTM* (ConvLSTM) di SOD pada *temporal state* di struktur internal *Saliency Generative Adversarial Networks* (SalGAN). Penambahan ConvLSTM pada SalGAN ini bertujuan untuk meningkatkan hasil prediksi GAN itu sendiri pada stabilitas temporal. Hal yang menjadi perbedaan mendasar antara CNN dengan ConvLSTM adalah lapisan CNN digunakan untuk mengekstraksi fitur statis di dalam input frames [7], sedangkan ConvLSTM digunakan untuk prediksi fiksasi atau obsesif objek yang bersifat sequential [8] Namun, sebenarnya CNN dan LSTM merupakan pasangan model yang kemampuannya saling melengkapi dalam untuk memodelkan *sequential data* atau *time-series* [9]. Maka dari itu, dikembangkanlah metode ConvLSTM.

Perbandingan utama antara LSTM dan ConvLSTM adalah ConvLSTM mengandalkan operasi konvolusi dibandingkan perkalian matriks seperti yang ada pada LSTM [10]. Dengan cara ini ConvLSTM dapat digunakan untuk membagun model stabilitas. Maka dari itu, penulis memilih ConvLSTM untuk mempelajari representasi dari SOD untuk mencapai stabilitas dalam iterasi temporal.

Namun, penerapan ConvLSTM ini mengandung algoritma yang cenderung bersifat kompleks dan membutuhkan *resource* komputasi berlebih mengingat adanya penambahan *memory cell* untuk mencegah *vanishing gradient* pada arsitektur RNN [11]. Oleh karena itu, di dalam penelitian ini, penulis turut menguji metode yang lebih sederhana untuk stabilitas temporal pada SOD dengan membalut lapisan convolutional dengan operasi *temporal Exponential Moving Average* (EMA) [12]. Nantinya, iterasi penggunaan *previous weight* yang dilakukan oleh EMA akan mempercepat proses konvergensi gradien [13], sehingga akan mengoptimisasi nilai *loss function* dan pada akhirnya mengurangi beban *resource* komputasi. Dengan demikian, penulis akan membandingkan hasil prediksi dari kedua pendekatan tersebut. Lalu, apabila peforma EMA menunjukkan hasil yang tidak signifikan atau lebih baik dari ConvLSTM, maka penulis akan mengusulkan penggunaan EMA yang jauh lebih sederhana pada SOD. Pada jurnal ini, penulis akan mendeteksi *object saliency* pada dataset UCF-Sports.

2 Previous Research

Penelitian SOD ini membutuhkan arsitektur *Recurrent Neural Network* (RNN) untuk menangani *sequential data* berupa video frame. Namun, RNN tradisional gagal menangkap evolusi jangka panjang, dan melatih RNN dengan jeda 5-10 menit terbukti sulit karena terjadi *vanishing gradient* dan *exploding gradient* [14]. Maka, untuk mengatasi masalah ini, jaringan LSTM diterapkan [15] dalam *temporal recurrences*. Dibandingkan dengan RNN konvensional, jaringan LSTM terbukti mampu menangkap dan menyimpan memori pada *sequential data* dalam rentang waktu yang lebih lama, sehingga menghasilkan akurasi prediksi yang lebih baik. Oleh karena itu, prediksi *object saliency* pada video diharapkan dapat mencapai peforma yang lebih efisien dengan menggunakan jaringan LSTM.

Di sisi lain, LSTM dan ConvLSTM merupakan pasangan model yang kinerjanya saling melengkapi [9]. LSTM memenuhi syarat untuk memodelkan *sequential data* secara temporal, sementara CNN mengurangi variasi frekuensi [9][16]. Oleh karena itu, ada beberapa cara untuk menggabungkan CNN, RNN (seperti, LSTM), dan *Deep Fully Connected Neural Networks* [17]. Jika metode ini digabungkan dalam framework yang sama, maka dapat meningkatkan peforma masing-masing model secara individu [16]. Performa ini juga telah diuji oleh penelitian yang dilakukan Rogério Luís de C. Costa[18]. Pada jurnalnya, peforma ConvLSTM berhasil dibuktikan lebih baik daripada LSTM karena hasil evaluasi pemodelannya menunjukkan bahwa ConvLSTM memiliki nilai RMSE yang signifikan lebih rendah daripada LSTM.

Di sisi lain, EMA dirancang untuk sensitif memberikan bobot lebih besar secara eksponensial pada data terbaru yang bersifat *real-time* karena dianggap memiliki relevansi yang tinggi. Sebagai dampaknya EMA akan merespon perubahan informasi data secara lebih cepat. Dalam dunia *neural network*, metode ini sudah diterapkan dalam penurunan gradien dengan momentum[13] untuk mempercepat konvergensi, dengan cara mengganti gradien saat ini dengan rata-rata pergerakan eksponensial dari gradien saat ini dan masa lalu yang berasal dari kumpulan data kecil.

Jurnal yang dirilis pada 2019 dengan judul *"Simple VS Complex Temporal Recurrences for Video Saliency Prediction"* merupakan penelitian pionir yang menggunakan modul EMA dalam arsitektur RNN. Penelitian ini muncul karena secara umum penambahan modul EMA pada arsitektur RNN cenderung cocok dilakukan untuk mencapai stabilitas temporal [1]. Pada penerapannya di RNN, operasi temporal EMA ini akan disematkan pada *convolutional layer*. Dengan menggunakan perulangan ini, output yang dihasilkan akan selalu menjadi *smoothed average* dari kondisi sebelumnya. Alhasil, penelitian ini menunjukkan bahwa pendekatan SalEMA di *temporal state* pada arsitektur SalGAN menghasilkan performa dengan signifikasi yang kecil dibandingkan dengan ConvLSTM untuk training video dataset bernama DHF1K. Dimana SalEMA menunjukkan nilai evaluasi AUC-J sebesar 0.890 dan s-AUC sebesar 0.667, sedangkan ConvLSTM memiliki nilai evaluasi AUC-J sebesar 0.887 dan s-AUC sebesar 0.693. Dengan demikian, penelitian tersebut berhasil menunjukkan bahwa pendekatan EMA yang lebih sederhana berhasil bersaing dengan ConvLSTM pada *saliency task*.

Oleh karena itu, jika EMA berhasil mencapai performa yang serupa atau lebih baik daripada LSTM, maka modifikasi yang lebih sederhana ini seharusnya lebih dipertimbangkan untuk dikembangkan daripada modul LSTM yang lebih kompleks. Bahkan, hasil dari kedua modifikasi arsitektur menunjukkan bahwa kedua model berhasil mengeluarkan *saliency map* yang hampir serupa pada dataset DHF1K. Oleh sebab itu, penulis ingin menguji performa dari kinerja penambahan modul ini pada dataset UCF-Sports.

3 Methodology

Salient object detection merupakan langkah awal untuk komputer mengenali objek (*object recognition*). Kemampuan ini diawali dengan adanya proses ekstraksi objek dari latar belakangnya sebelum sistem kecerdasan merekognisi atau mengenali objek tersebut [19]. Lalu, pertanyaan berikutnya adalah bagaimana sistem dalam suatu mesin komputer dapat mengekstrak area objek yang secara visual manusia terlihat menonjol dari latar belakangnya? Untuk mewujudkannya, para peneliti akan membangun arsitektur dengan algoritma *supervised deep learning* yang dapat mengestimasi heatmap dari probabilitas kemungkinan piksel yang menarik perhatian manusia [19]. Cara ini diimplementasikan mengingat ada banyak objek yang akan dimasukkan ke dalam mesin komputer. Akibatnya, komputer akan menerima informasi objek yang berlebihan (*overload*). Maka dari itu, *deep learning* harus menghasilkan heatmap yang terdiri dari persebaran warna sesuai dengan nilai probabilitas dari fokus objek. Heatmap ini akan membantu komputer untuk menentukan bagian-bagian penting dalam suatu gambar, sehingga heatmap bisa disebut juga sebagai *saliency map*. Pada akhirnya, hasil analisis ini dapat membantu manusia untuk mendapatkan informasi suatu objek penting pada bidang visual yang luas dan sulit dijangkau oleh indra penglihatan manusia sendiri.

3.1 Proposed Architecture

Untuk melakukan prediksi video saliensi pada dataset tersebut, penulis menggunakan skema encoder-decoder. Topologi ini diadopsi dari SalGAN [20], dimana encoder bertanggung jawab untuk klasifikasi gambar menggunakan arsitektur VGG-16 dan decoder menggunakan layer yang sama dengan encoder namun dengan urutan terbalik dan diselingi dengan upsampling. Pada arsitektur ini, penulis melatih model SalGAN dengan metode Binary Cross Entropy. Pada arsitektur ini, penulis memperkenalkan *Temporally Aware Component* berupa penambahan layer ConvLSTM atau EMA di antara bagian encoder dan decoder. Penambahan layer ini dimaksudkan agar komputer dapat mempelajari input dengan lebih baik sehingga akurasi prediksinya dapat meningkat. Arsitektur model ini dapat digambarkan sebagai berikut:

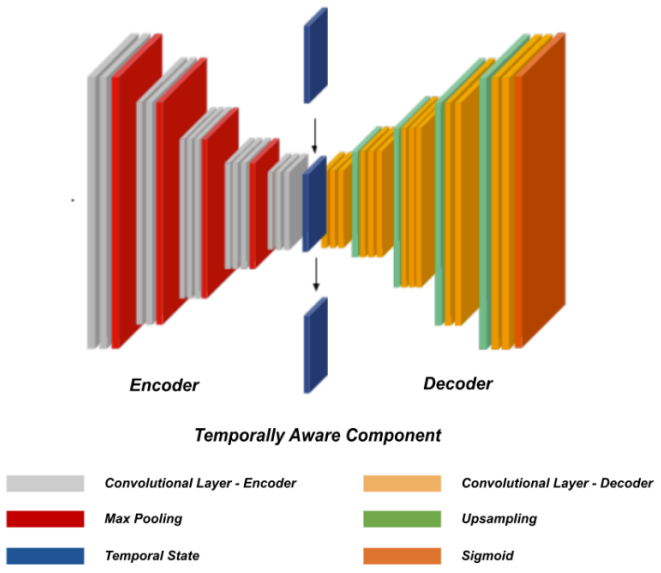

Figure 1: Arsitektur Model

3.2 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) adalah salah satu jenis dari arsitektur *Recurrent Neural Network* (RNN) yang digunakan untuk mengatasi *vanishing gradient* pada RNN ketika memproses *data sequential* yang panjang. Karena permasalahan ini akan mengakibatkan RNN gagal dalam menangkap *long term dependencies*[21], sehingga mengurangi akurasi dari suatu prediksi pada RNN [22]). LSTM memiliki penambahan *memory cell* untuk menyimpan pola-pola informasi pada data dalam periode masa lalu ataupun dari waktu ke waktu. Setiap neuron LSTM memiliki beberapa gates yang mengatur memori dalam setiap neuron itu sendiri, yaitu *Forget Gate*, *Input Gate*, dan *Output Gate* (Figure 2). Berkat hal itu, LSTM dapat mempelajari dan "mengingat" berbagai data, baik yang akan disimpan ataupun yang nantinya dibuang. Oleh karena kemampuannya yang dapat mengingat input data sebelumnya sebelumnya, maka LSTM seringkali digunakan untuk pemrosesan data teks, video, dan data *time series*.

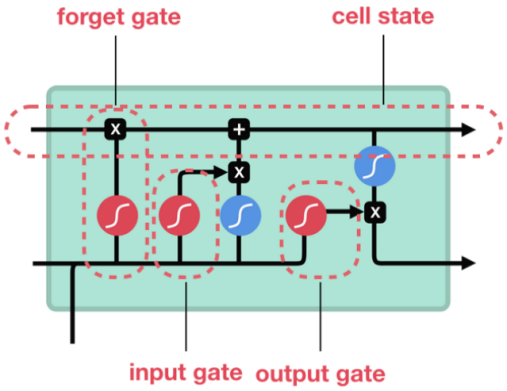

Figure 2: Long Short Term Memory (LSTM)

Dalam bidang *computer vision*, model LSTM biasanya akan mampu memurnikan suatu objek visual, sehingga *neural network* dapat "melihat" dan menilai objek secara lebih jelas. Namun, optimalisasi kinerja model ini sangat bergantung pada komponen waktu. Jadi sebenarnya, pendekatan ini tidak dapat langsung digunakan untuk memprediksi *object saliency*, karena LSTM bekerja pada urutan vektor yang bervariasi waktu [23]. Maka dari itu, penambahan modul LSTM sebagian besar hanya terjadi di dalam penelitian dan analisis *data sequential*.

3.3 Convolutional Long Short Term Memory (ConvLSTM)

Convolutional LSTM (ConvLSTM) merupakan modifikasi dari LSTM, dimana ConvLSTM didesain untuk menerima input data 3 dimensi, sedangkan LSTM digunakan untuk menerima input data 1 dimensi. Vector x_t, h_t, dan C_t diubah menjadi bentuk 3D tensor [24]. Sedangkan, matriks weight w untuk setiap gerbang ConvLSTM diganti dengan filter konvolusi. Penggantian ini menyebabkan operasi perkalian matriks yang terjadi pada setiap gerbang digantikan dengan operasi konvolusi seperti yang ditunjukkan pada Figure 3 [25]. Digunakannya operasi konvolusi memungkinkan ConvLSTM untuk menangkap fitur-fitur spasial yang ada pada data multidimensi sehingga ConvLSTM dapat menghasilkan *spatial feature map* yang lebih baik [26]. Sama seperti LSTM, ConvLSTM juga memiliki *cell state* yang membawa memori informasi dari sel-sel sebelumnya. ConvLSTM juga akan menentukan kondisi selanjutnya dari sel berdasarkan input dan hidden state sel sebelumnya.

Input x_t dan data dari hidden state sebelumnya h_{t-1} akan masuk ke forget gate, dimana pada forget gate ini akan ditentukan apakah informasi yang dibawa oleh C_{t-1} penting untuk diteruskan atau tidak. Dengan weight W_{xf} dan bias b_f, input x_t dan data dari hidden state sebelumnya h_{t-1} akan diolah menggunakan fungsi sigmoid, dimana hasilnya f_t akan berada pada interval 0 dan 1. Hasil ini kemudian akan dikalikan dengan cell state sebelumnya (C_{t-1}). Perkalian ini merupakan perkalian antar elemen, dimana artinya dalam cell state yang sama, sangat memungkinkan jika ada informasi yang dibuang dan ada informasi lain yang tetap disimpan. Jika hasil perkaliannya bernilai 0, maka informasi tersebut tidak dianggap penting untuk dilanjutkan dan akan dilupakan (*forget*) atau dibuang. Namun apabila output fungsi sigmoidnya bernilai 1, maka informasi tersebut masih dianggap penting dan akan diteruskan ke cell state berikutnya.

Input x_t dan data dari hidden state sebelumnya h_{t-1} juga akan masuk ke input gate dengan weight W_{xi} dan bias b_i. Pada input gate, terdapat fungsi tanh dimana data akan diolah menjadi kandidat informasi baru yang akan ditambahkan pada cell state C_{t-1}. Namun sebelum terjadi penambahan pada C_{t-1}, kandidat informasi ini akan dikalikan dengan

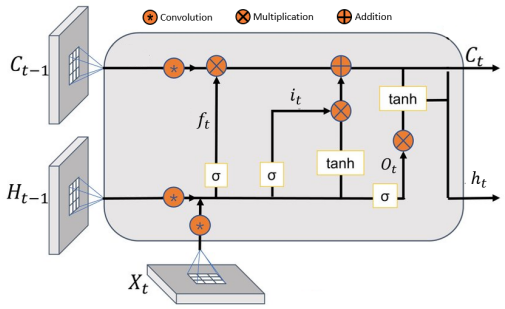

Figure 3: Convolutional LSTM (ConvLSTM)

nilai dari fungsi sigmoid pada input gate untuk menentukan apakah kandidat informasi baru tersebut penting untuk ditambahkan pada cell state C_{t-1} atau tidak. Apabila output fungsi sigmoid input gate bernilai 0, maka hasil perkalian output fungsi sigmoid dengan fungsi tanh juga akan bernilai 0, dimana artinya kandidat informasi tersebut dianggap tidak penting dan tidak akan ditambahkan pada cell state. Namun sebaliknya, apabila output fungsi sigmoidnya bernilai 1, maka hasil perkalian output fungsi sigmoid dengan fungsi tanh akan bernilai sama dengan output fungsi tanh, dimana artinya kandidat informasi tersebut dianggap penting dan akan ditambahkan pada cell state.

Cell state yang nilainya telah diperbaharui dengan f_t dan i_t kemudian akan diteruskan ke sel ConvLSTM selanjutnya. Selain itu, nilai cell state C_t ini juga akan diolah dengan menggunakan fungsi tanh, yang kemudian dikalikan dengan output dari fungsi sigmoid pada output gate dengan memperhatikan weight W_{xo} dan bias b_o. Data yang diolah pada fungsi sigmoid output gate berasal dari input x_t dan data dari hidden state sebelumnya h_{t-1}. Pada output gate ini akan ditentukan seberapa banyak output yang akan disimpan dan dijadikan data untuk hidden state selanjutnya. Hasil o_t inilah yang kemudian akan diteruskan sebagai hidden state untuk sel berikutnya.

Perhitungan yang terjadi pada masing-masing gerbang ConvLSTM dapat dituliskan secara matematis sebagai berikut [8], dimana '$*$' melambangkan operasi konvolusi dan '$\circ$' melambangkan *Hadamard product*:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ C_{t-1} + b_i) \tag{1}$$
$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \circ C_{t-1} + b_f) \tag{2}$$
$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ C_{t-1} + b_o) \tag{3}$$

Nilai dari cell state dan hidden state yang baru kemudian dihitung menggunakan rumus sebagai berikut:

$$C_t = f_t \circ C_{t-1} + i_t \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \tag{4}$$
$$h_t = o_t \circ \tanh(C_t) \tag{5}$$

Penggunaan arsitektur ConvLSTM ini diharapkan agar input dapat di-*encode* menjadi sebuah frame pada waktu t dan outputnya dapat diberikan pada bagian decoder untuk menghasilkan *saliency map*. Video frame dimasukkan pada model secara berkala, kemudian model menghasilkan urutan *saliency map* yang berkorelasi dengan waktu.

3.4 Exponential Moving Average (EMA)

Sebagai pendekatan alternatif, *Exponential Moving Average* (EMA) digunakan pada sebuah layer spesifik sehingga *convolutional state* pada saat t (S_t) akan terjadi peluruhan *weighted average* secara eksponensial dari *state* saat ini dan seluruh *state* sebelumnya [1]. Output EMA_t kemudian akan disebarkan lebih lanjut dalam model. Perlu dicatat bahwa terdapat *hyperparameter* (α) yang mempengaruhi dampak pada *state* sebelum *state* pada waktu t, dimana nilai yang semakin kecil akan memperbesar dampaknya.

$$EMA_t = \alpha S_t + (1 - \alpha) EMA_{t-1} \tag{6}$$

Dengan diterapkannya EMA pada *temporal state*, akan didapatkan hasil optimasi yang lebih baik pada model awal, yaitu SalGAN. Berkurangnya bobot akan berdampak pada berkurangnya nilai *loss function* ketika dilakukannya *training*. Hal tersebut akan berpengaruh terhadap konvergensi, dimana nilai *loss function* yang semakin rendah menandakan akan mendekatnya gradien kepada konvergensi.

Selain itu, rekurensi ini dapat diimplementasikan dengan mudah, dibandingkan dengan ConvLSTM. Hal tersebut dikarenakan *resource* komputasi yang digunakan oleh EMA lebih efisien. Tidak banyak daya komputasi yang dibutuhkan dalam kalkulasi EMA. Penulis akan menempatkan fungsi EMA pada beberapa layer yang berbeda dengan $\alpha = 0.1$, yang kemudian model ini disebut dengan SalEMA.

4 Experiment

4.1 Dataset

Di dalam penelitian ini, dataset yang digunakan adalah dataset UCF-Sports, dimana dataset ini berisikan kumpulan adegan dari berbagai jenis olahraga yang seringkali disiarkan di siaran televisi, seperti BBC dan ESPN. Adegan-adegan ini ditampilkan dalam 150 sequence video dari beragam sudut pandang dengan resolusi 720x480 yang semuanya didapat dari beragam situs *footage*, misalnya BBC Motion Gallery dan GettyImages. Dari 150 sequence video, sebanyak 103 video merupakan bagian dari data *training*, dan sisanya merupakan data *testing*. Dataset ini digunakan karena diharapkan deteksi salient object yang dilakukan tidak hanya dapat menemukan warna yang paling berbeda dibandingkan backgroundnya, namun juga komputer diharapkan dapat menangkap informasi penting dari suatu rangkaian kejadian (*action recognition*) sehingga tidak terjadi misinterpretasi data [27]. Beragamnya tipe gambar pada video frame dalam UCF Sport, seperti jarak jauh dekatnya foto *salient object* serta perbedaan warna grayscale dan RGB, juga membuat dataset ini mampu melatih model untuk menentukan *salient object* dengan lebih baik. Selain itu, ukuran dataset UCF-Sports ini relatif lebih kecil apabila dibandingkan dengan dataset yang berisi *video sequence* lainnya (misalnya DHF1K).

4.2 Training

Seperti yang telah disebutkan pada section sebelumnya, dalam penelitian ini, akan diterapkan dua pendekatan berbeda, yaitu menggunakan ConvLSTM dan EMA. Kemudian, akan dilakukan proses training terhadap masing-masing model menggunakan dataset UCF-Sports.

Pada pendekatan EMA, model akan dilatih menggunakan *pre-trained model* yang dihasilkan penelitian milik Linardos, yaitu SalEMA30 [1]. Hasil dari pelatihan tersebut kemudian disebut model SalEMA. Model SalEMA dilatih selama 10 epochs menggunakan tingkat pembelajaran sebesar 0.0000001. Berbeda dengan pendekatan EMA, model dengan pendekatan ConvLSTM akan dilatih dari awal, yang kemudian disebut dengan CLSTM30. Jumlah epoch dan tingkat pembelajaran yang diterapkan pada model ini memiliki nilai yang sama dengan yang diterapkan pada model EMA.

4.3 Result and Evaluation

Berdasarkan pelatihan yang dilakukan terhadap masing-masing model, dihitung nilai fungsi loss menggunakan Binary-Cross Entropy (BCE). Nilai fungsi loss yang didapat dari setiap epoch dijumlahkan, dan dikalkulasikan rata-ratanya. Nilai rata-rata dari fungsi loss hasil pelatihan model CLSTM30 dibandingkan dengan nilai yang didapatkan dari model SalEMA. Hasil dari kedua nilai rata-rata fungsi loss akan dibandingkan di section berikutnya. Kemudian, dilakukan pengujian terhadap kedua model menggunakan data testing dari dataset yang sama. Hasil dari pengujian kedua model tersebut adalah berupa gambar-gambar prediksi salient object pada setiap sequence video. Gambar prediksi ini berwarnakan hitam-putih, dimana warna putih menandakan salient object yang terdeteksi.

4.3.1 Model Comparison

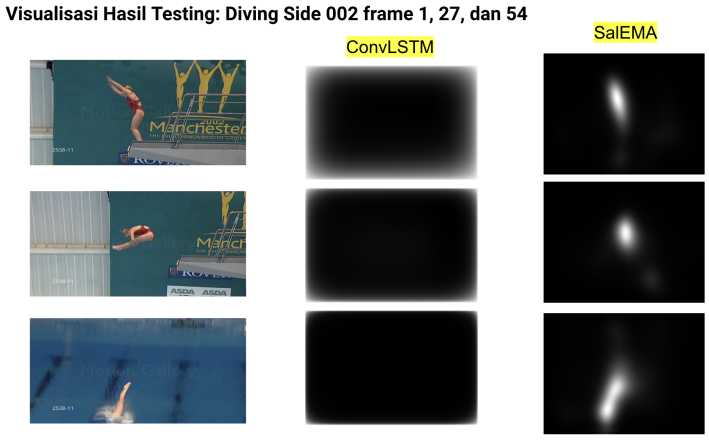

Figure 4: Visualisasi Hasil Testing

Table 1: Model Performance Comparison

Metode	Nilai Rata-Rata Loss Function
ConvLSTM	3.64472584724426273
SalEMA	0.7709852159023285

Berdasarkan tabel diatas, dapat dilihat bahwa nilai rata-rata fungsi loss pada model SalEMA bernilai jauh lebih kecil dibandingkan dengan nilai yang didapat oleh model CLSTM30. Nilai yang jauh lebih kecil ini menandakan bahwa training yang dilakukan pada model SalEMA dapat memberikan hasil yang lebih baik dibandingkan CLSTM30.

Kemudian, peneliti mengambil sample dari gambar prediksi milik CLSTM30 (atau ConvLSTM) dan membandingkannya dengan gambar asli pada salah satu sequence video, yaitu Diving-Side-002. Gambar yang diambil merupakan gambar pertama (frame 1), gambar ke-27, dan gambar terakhir (frame 54). Hasil yang didapat pada Figure 4 memperlihatkan bahwa model SalEMA dapat memprediksi salient object dengan baik. Di sisi lain, CLSTM30 sama sekali tidak dapat memprediksi saliency yang ada pada gambar.

5 Conclusion

Berbagai metode untuk *saliency task* yang telah dikembangkan pada penelitian sebelumnya cenderung memiliki tingkat kompleksitas yang tinggi. Dalam *paper* ini, dilakukan perbandingan pendekatan ConvLSTM dengan EMA, dimana pendekatan dengan metode EMA memiliki tingkat kompleksitas yang rendah karena tidak menggunakan sel memori dan hanya mentransformasi nilai bobot menjadi lebih rendah. Setelah dilakukan percobaan, ternyata kedua metode ini memberikan perbedaan hasil yang signifikan. Nilai *loss function training* ConvLSTM mencapai 3.64, sedangkan nilai *loss function training* EMA hanya sebesar 0.77. Hal ini membuktikan bahwa EMA memiliki performa yang jauh lebih baik dibandingkan ConvLSTM dalam *salient object detection* pada dataset UCF-Sport. Dengan demikian, pendekatan EMA layak dipertimbangkan untuk digunakan dan dikembangkan lebih lanjut dalam melakukan salient object detection task.

Di sisi lain, dikarenakan pada penelitian ini, model ConvLSTM belum dapat memprediksi salient object, maka disarankan untuk dapat menyempurnakan kembali model dengan pendekatan ConvLSTM dengan mulai memperbanyak epoch pada pelatihan model untuk penelitian selanjutnya.

References

[1] Panagiotis Linardos, Eva Mohedano, Juan José Nieto, Noel E. O'Connor, Xavier Giró-i-Nieto, and Kevin McGuinness. Simple vs complex temporal recurrences for video saliency prediction. *CoRR*, abs/1907.01869, 2019. URL `http://arxiv.org/abs/1907.01869`.

[2] Yo Umeki, Isana Funahashi, Taichi Yoshida, and Masahiro Iwahashi. Salient object detection with importance degree. *IEEE Access*, 8:147059–147069, 2020. doi:10.1109/ACCESS.2020.3014886.

[3] Wenguan Wang, Jianbing Shen, Jianwen Xie, Ming-Ming Cheng, Haibin Ling, and Ali Borji. Revisiting video saliency prediction in the deep learning era. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43 (1):220–237, 2021. doi:10.1109/TPAMI.2019.2924417.

[4] Xuelong Li, Dawei Song, and Yongsheng Dong. Hierarchical feature fusion network for salient object detection. *IEEE Transactions on Image Processing*, 29:9165–9175, 2020. doi:10.1109/TIP.2020.3023774.

[5] Lai Jiang, Mai Xu, Tie Liu, Minglang Qiao, and Zulin Wang. Deepvs: A deep learning based video saliency prediction approach. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 625–642, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01264-9.

[6] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. *CoRR*, abs/1801.07424, 2018. URL `http://arxiv.org/abs/1801.07424`.

[7] Xiankai Lu, Chao Ma, Jianbing Shen, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep object tracking with shrinkage loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2386–2401, 2022. doi:10.1109/TPAMI.2020.3041332.

[8] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015. URL `http://arxiv.org/abs/1506.04214`.

[9] Ning Xue, Isaac Triguero, Grazziela P. Figueredo, and Dario Landa-Silva. Evolving deep cnn-lstms for inventory time series prediction. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 1517–1524, 2019. doi:10.1109/CEC.2019.8789957.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[11] Dilantha Haputhanthri and Adeesha Wijayasiri. Short-term traffic forecasting using lstm-based deep learning models. In *2021 Moratuwa Engineering Research Conference (MERCon)*, pages 602–607, 2021. doi:10.1109/MERCon52712.2021.9525670.

[12] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. ISSN 0041-5553. doi:https://doi.org/10.1016/0041-5553(64)90137-5. URL https://www.sciencedirect.com/science/article/pii/0041555364901375.

[13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/sutskever13.html.

[14] Shuai Li, Wanqing Li, Chris Cook, Yanbo Gao, and Ce Zhu. Deep independently recurrent neural network (indrnn). *CoRR*, abs/1910.06251, 2019. URL http://arxiv.org/abs/1910.06251.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

[16] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584, 2015. doi:10.1109/ICASSP.2015.7178838.

[17] Li Deng and John Platt. Ensemble deep learning for speech recognition. In *Proc. Interspeech*, September 2014. URL https://www.microsoft.com/en-us/research/publication/ensemble-deep-learning-for-speech-recognition/.

[18] Rogério Luís de C. Costa. Convolutional-lstm networks and generalization in forecasting of household photovoltaic generation. *Engineering Applications of Artificial Intelligence*, 116:105458, 2022. ISSN 0952-1976. doi:https://doi.org/10.1016/j.engappai.2022.105458. URL https://www.sciencedirect.com/science/article/pii/S0952197622004481.

[19] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. doi:10.1109/CVPR.2007.383267.

[20] Junting Pan, Cristian Canton-Ferrer, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giró-i-Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *CoRR*, abs/1701.01081, 2017. URL http://arxiv.org/abs/1701.01081.

[21] Navin Kumar Manaswi. *RNN and LSTM*, pages 115–126. Apress, Berkeley, CA, 2018. ISBN 978-1-4842-3516-4. doi:10.1007/978-1-4842-3516-4_9. URL https://doi.org/10.1007/978-1-4842-3516-4_9.

[22] Zheng Zhao, Weihai Chen, Xingming Wu, Peter C. Y. Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017. doi:https://doi.org/10.1049/iet-its.2016.0208. URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2016.0208.

[23] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *CoRR*, abs/1611.09571, 2016. URL http://arxiv.org/abs/1611.09571.

[24] Nesma M. Rezk, Madhura Purnaprajna, Tomas Nordström, and Zain Ul-Abdin. Recurrent neural networks: An embedded computing perspective. *IEEE Access*, 8:57967–57996, 2020. doi:10.1109/ACCESS.2020.2982416.

[25] Changjiang Shi, Zhijie Zhang, Wanchang Zhang, Chuanrong Zhang, and Qiang Xu. Learning multiscale temporal–spatial–spectral features via a multipath convolutional lstm neural network for change detection with hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 05 2022. doi:10.1109/TGRS.2022.3176642.

[26] Jefferson Ryan Medel and Andreas E. Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *CoRR*, abs/1612.00390, 2016. URL http://arxiv.org/abs/1612.00390.

[27] Khurram Soomro and Amir Roshan Zamir. Action recognition in realistic sports videos. 2014.