ASSESSMENT OF LEARNING

Membandingkan Hasil Prediksi *Object Saliency* pada Video dengan Pendekatan *Convolutional Long Short Term Memory* (LSTM) dan *Exponential Moving Average* (EMA) dalam Arsitektur *Recurrent Neural Network* (RNN)

BERNADETHA | DIANA | FELICIA

Introduction

- Apa itu object saliency?
- Pentingnya mendeteksi object saliency
- Bagaimana saliency detection terjadi pada computer vision?

Object Saliency

? *Object Saliency*

- Suatu bagian objek tertentu yang memiliki informasi lebih bermakna dan berkesan dibandingkan bagian-bagian lainnya.
- Terlihat lebih menonjol dan menarik perhatian visual manusia.
- Hasil dari keterbatasan manusia dalam memfokuskan daya persepsi dan kognitif ketika melihat suatu objek.

Dampak positif:

Efisien dalam mengekstraksi suatu objek penting

Diterapkan dalam:
foreground extraction, *visual tracking*, *scene classification*, *semantic segmentation*, *video summarization*, dan *image retrieval*.

Deteksi *Object Saliency* dalam Computer Vision

Ekstraksi objek dari latar belakangnya sebelum sistem kecerdasan metakognisi/ mengenali objek tersebut.

Dengan cara:
mengestimasi ***heatmap* dari probabilitas piksel** yang menarik perhatian manusia.

Heatmap akan membantu komputer untuk mengambil bagian-bagian penting dalam suatu gambar **(=) *saliency map*.**

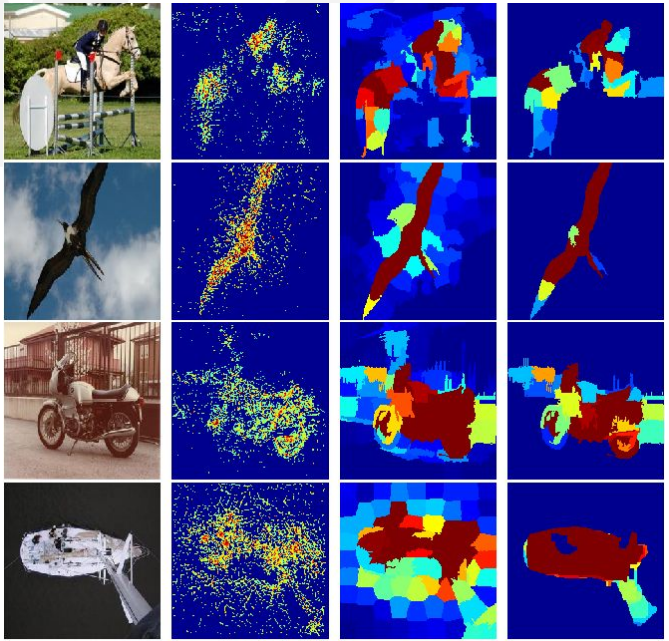
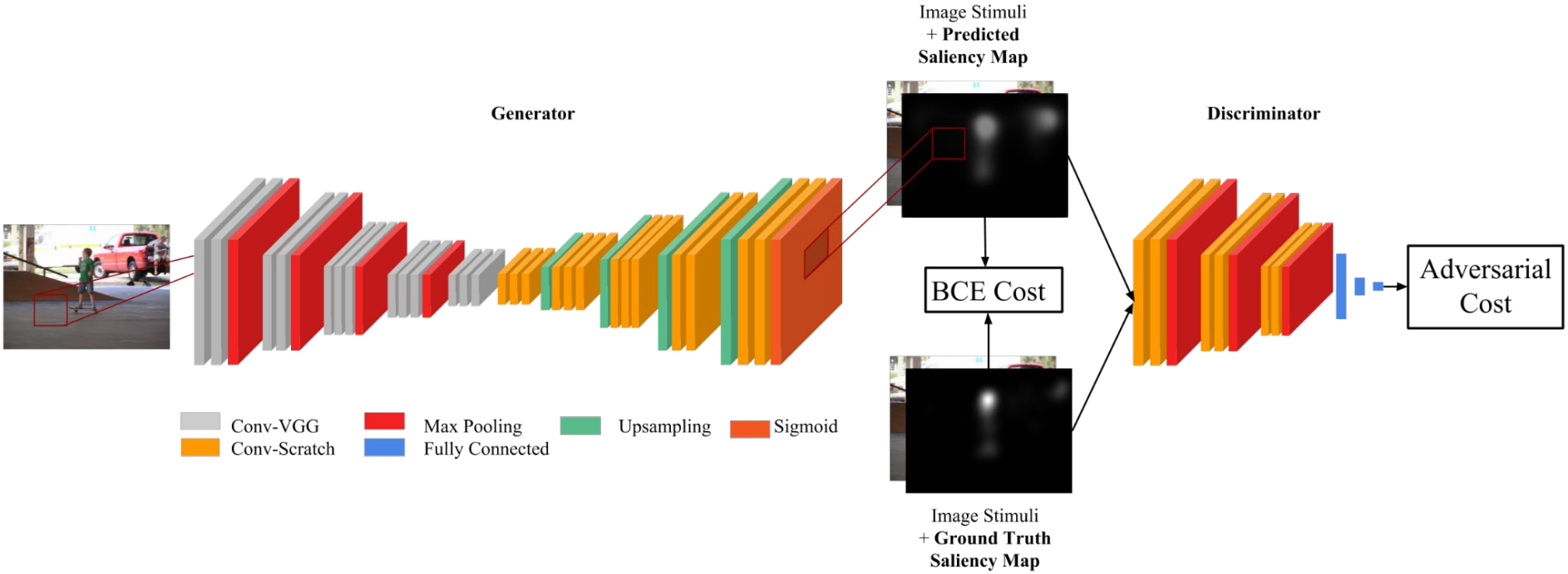
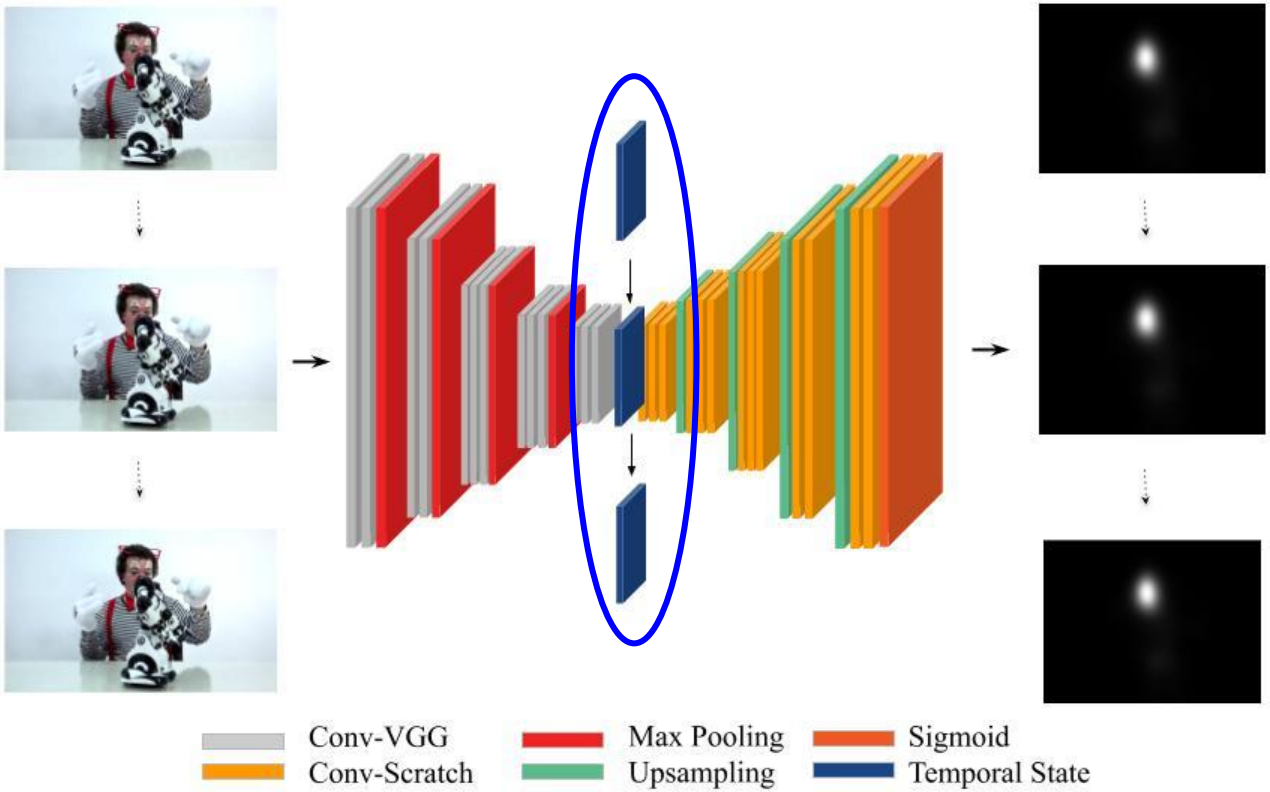


Figure 2. From left to right: original images, raw saliency maps,

Arsitektur Asli: SalGAN

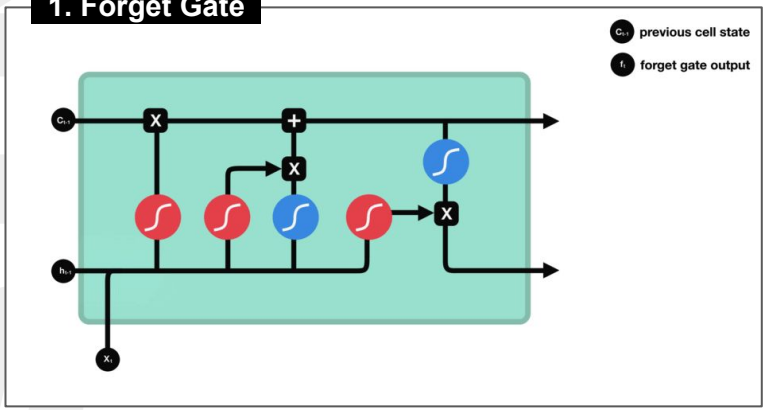


Arsitektur **ConvLSTM dan EMA**

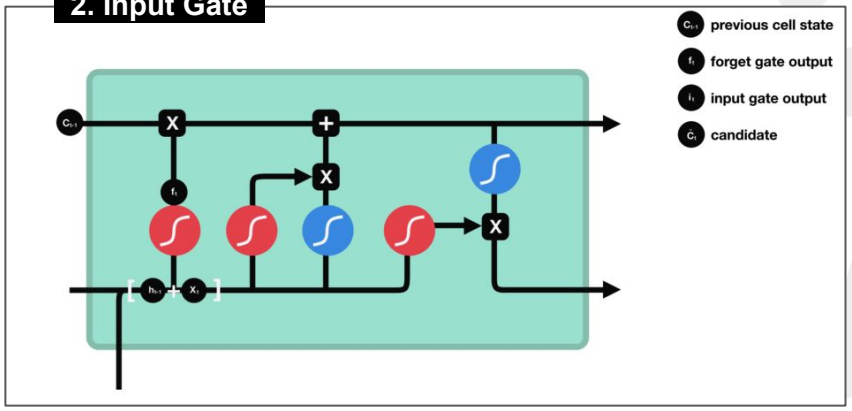


Convolutional Long Short Term Memory

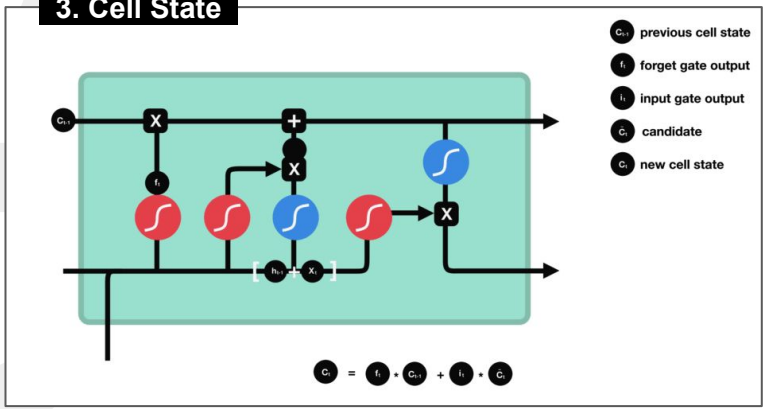
1. Forget Gate



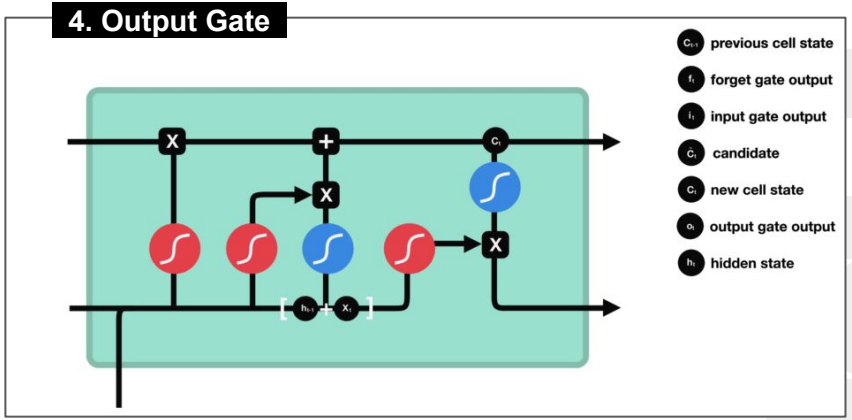
2. Input Gate



3. Cell State



4. Output Gate



Exponential Moving Average

Melakukan transformasi weight dari perhitungan di VGG16 pada proses Encoder dengan persamaan:

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

*EMA gives **exponentially decreasing weights** to past values.*

Pengaruh: **optimisasi**

- Meminimalisir nilai loss function
- Resource komputasi yang digunakan juga efisien.

```
    def forward(self, input_, prev_state=None):
        x = self.salgan[:self.ema_loc](input_)
        residual = x
        batch_size = x.data.size()[0]
        spatial_size = x.data.size()[2:]

        if self.dropout == True:
            x = dropout2d(x)
        # salgan[self.ema_loc] will act as the temporal state
        if prev_state is None:
            current_state = self.salgan[self.ema_loc](x) #Initially don't apply alpha as there is no
prev state we will consistently have bad saliency maps at the start if we were to do so.
        else:
            current_state = sigmoid(self.alpha)*self.salgan[self.ema_loc](x)+(1-
sigmoid(self.alpha))*prev_state
            #current_state = (self.alpha)*self.salgan[self.ema_loc](x)+(1-(self.alpha))*prev_state

        if self.residual == True:
            x = current_state+residual
        else:
            x = current_state

        if self.ema_loc < len(self.salgan)-1:
            x = self.salgan[self.ema_loc+1:](x)

        return current_state, x #x is a saliency map at this point
```

UCF-Sport

150 sequence video 720x480

DATA

Hasil Training

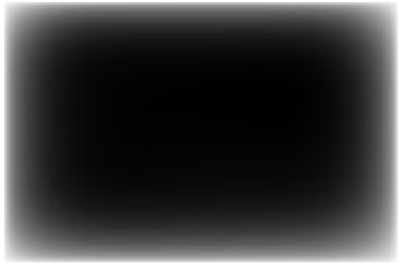
Metode	Nilai rata-rata loss function*
ConvLSTM	3.6447258472442627 3
SalEMA	0.7709852159023285

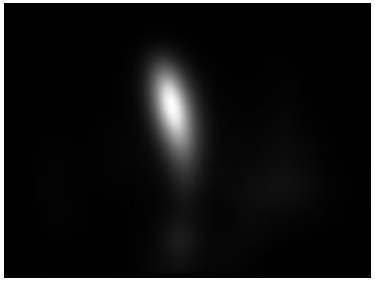
**untuk training 10 epochs*

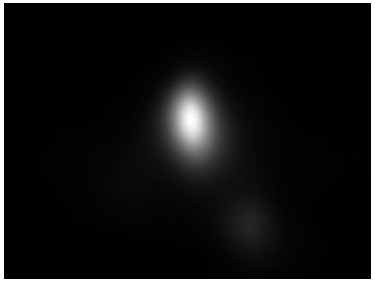
Visualisasi Hasil Testing: Diving Side 002 frame 1, 27, dan 54

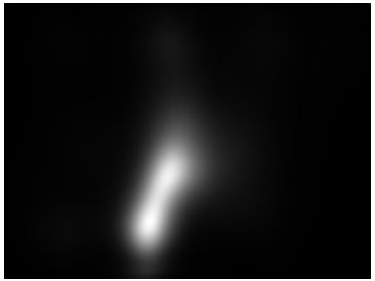
ConvLSTM

SalEMA





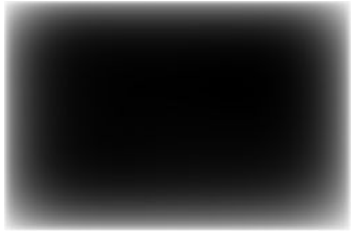


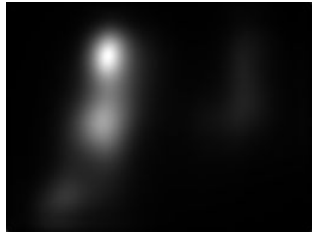


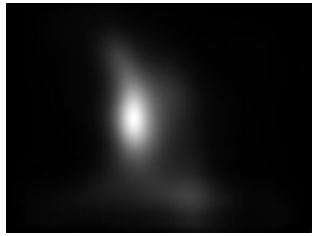
Visualisasi Hasil Testing

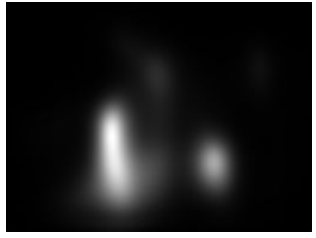
ConvLSTM

SalEMA









Kesimpulan

- Prediksi *Object Saliency* dengan pendekatan EMA menunjukkan nilai *loss function training* yang lebih rendah daripada ConvLSTM.
- EMA lebih dipertimbangkan untuk dikembangkan di dalam prediksi *object saliency* daripada modul ConvLSTM yang lebih kompleks.

THANK YOU