

Efficient Visual Attention with Deep Learning

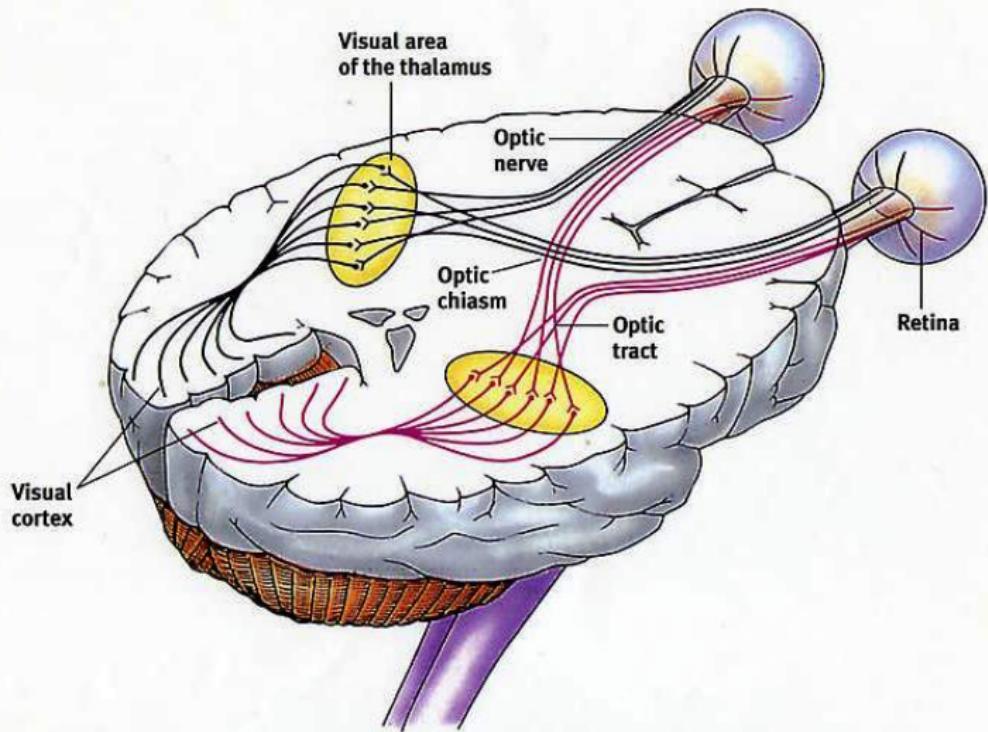
Erik Perillo, Esther Colombini

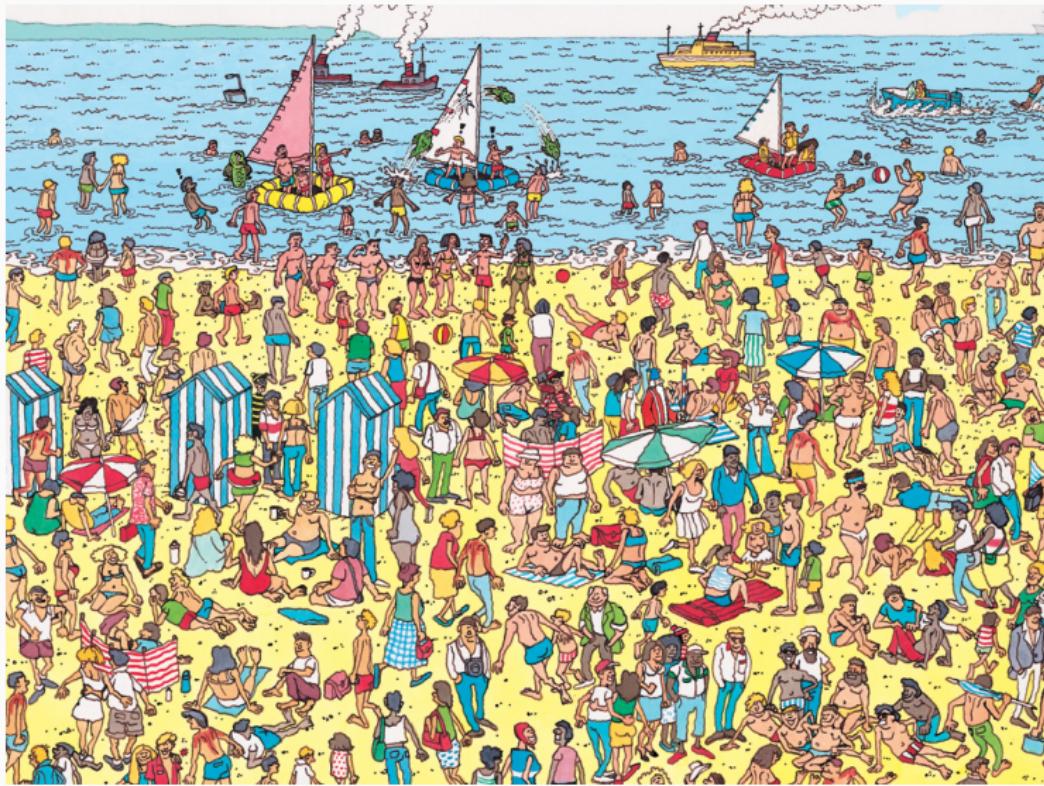
Instituto de Computação – Unicamp

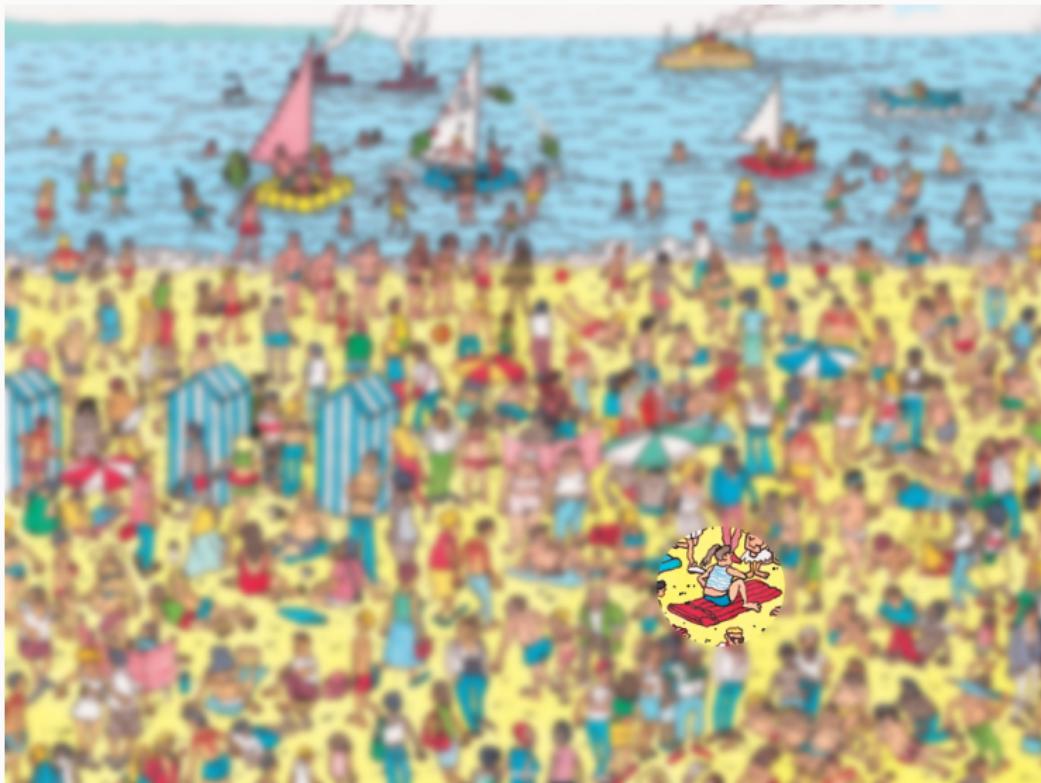
Visão

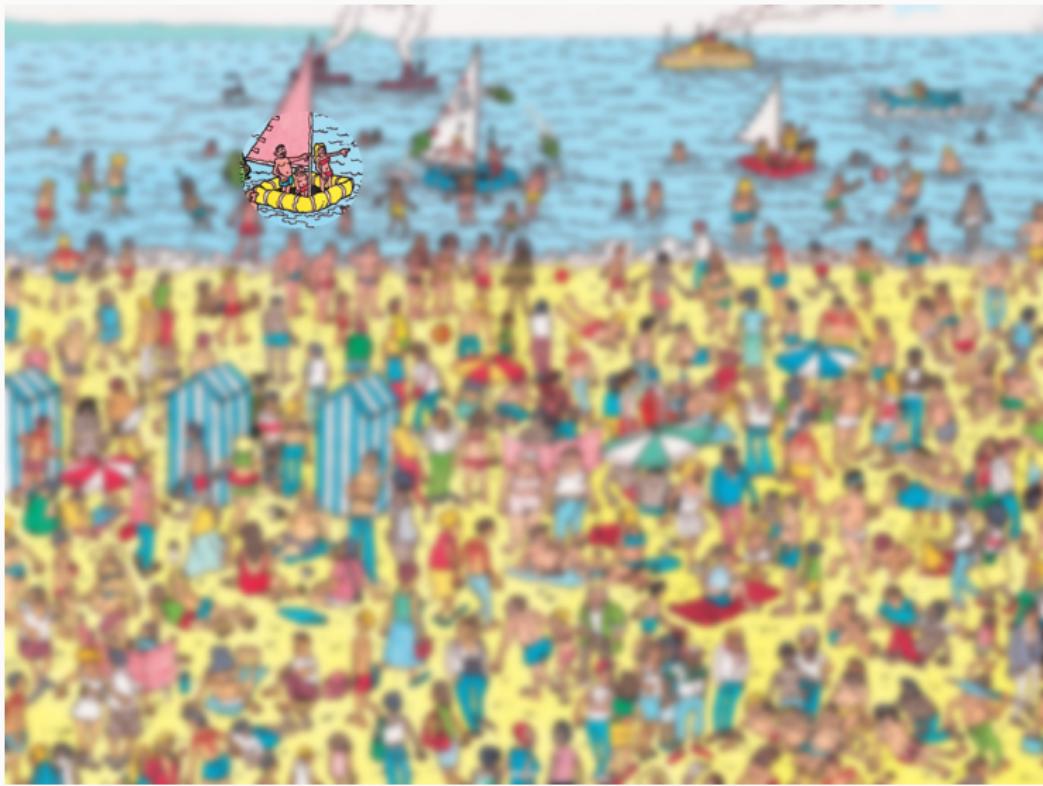












Atenção Visual

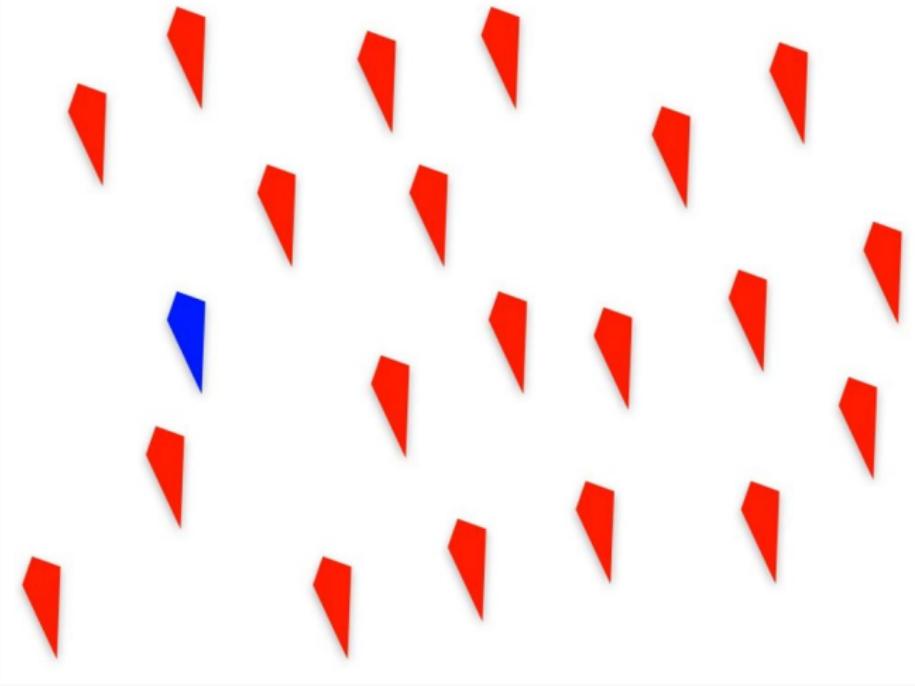
Seleção de uma certa região espacial do campo visual para posterior processamento cognitivo

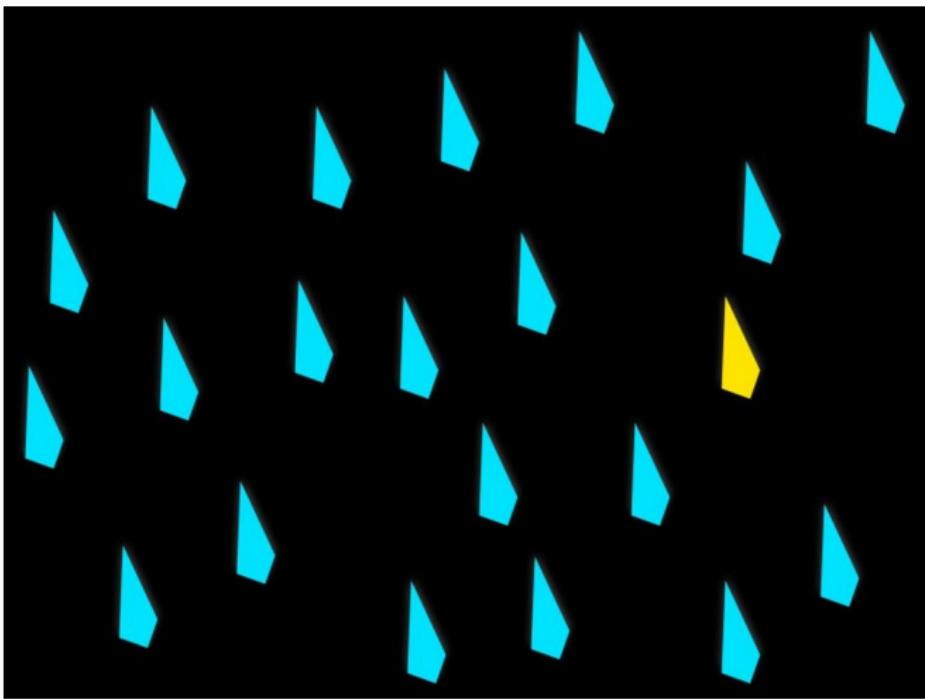
Atenção: Top-down versus Bottom-up

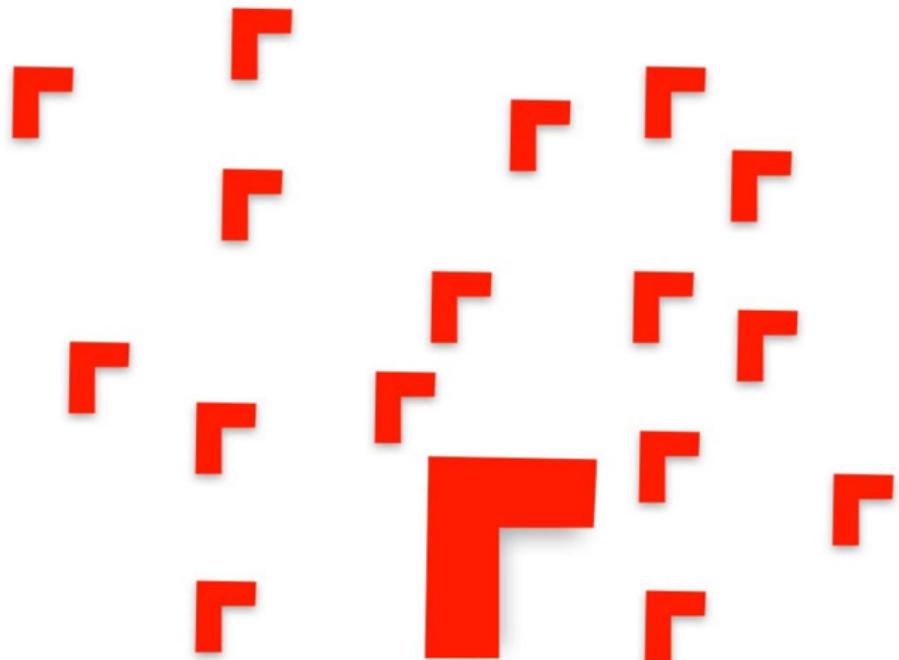
- Top-down: Estímulo interno do ser que direciona a atenção a padrões específicos de estímulo visual

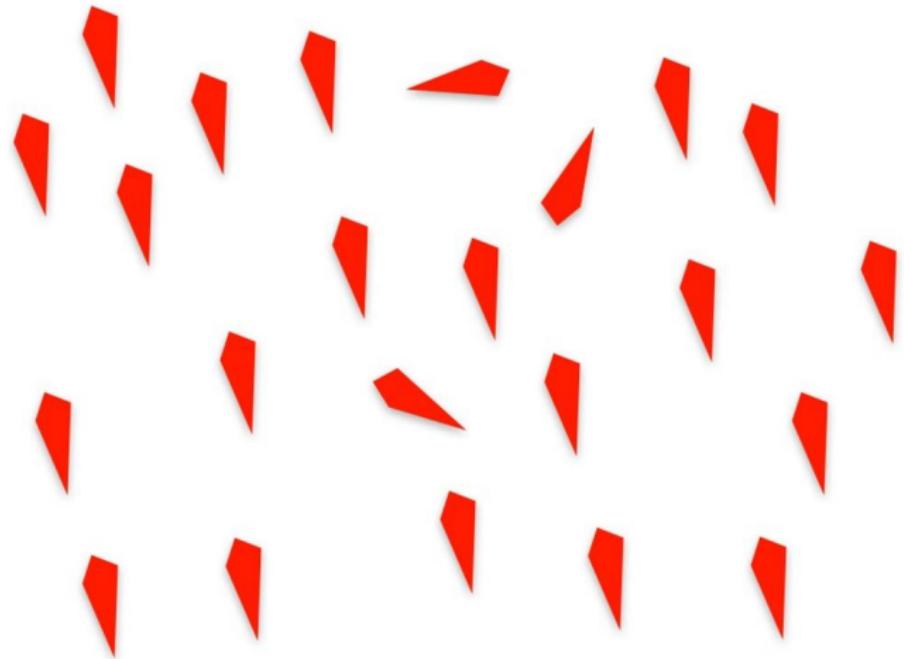
Atenção: Top-down versus Bottom-up

- Top-down: Estímulo interno do ser que direciona a atenção a padrões específicos de estímulo visual
- **Bottom-up (saliência visual):** Estímulo externo que capta a atenção visual







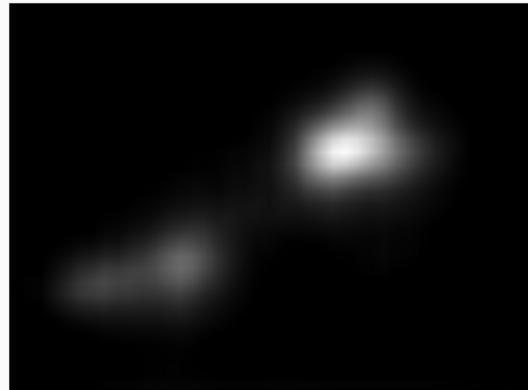


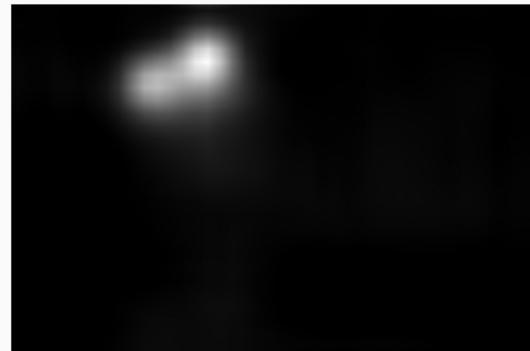


Mapa de saliência









Podemos fazer um computador identificar saliências visuais?

Modelo de saliência visual

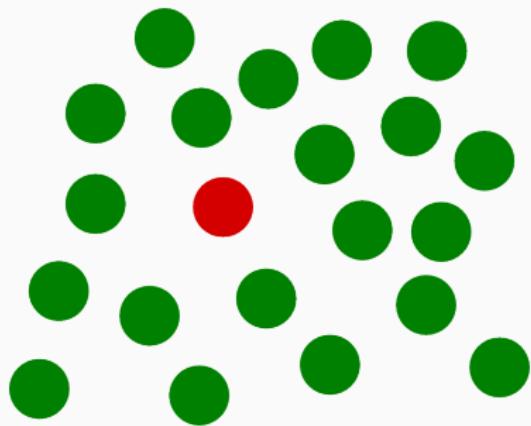
Ideia:

- Dada uma imagem, gerar um mapa de saliência coerente com o que humanos gerariam

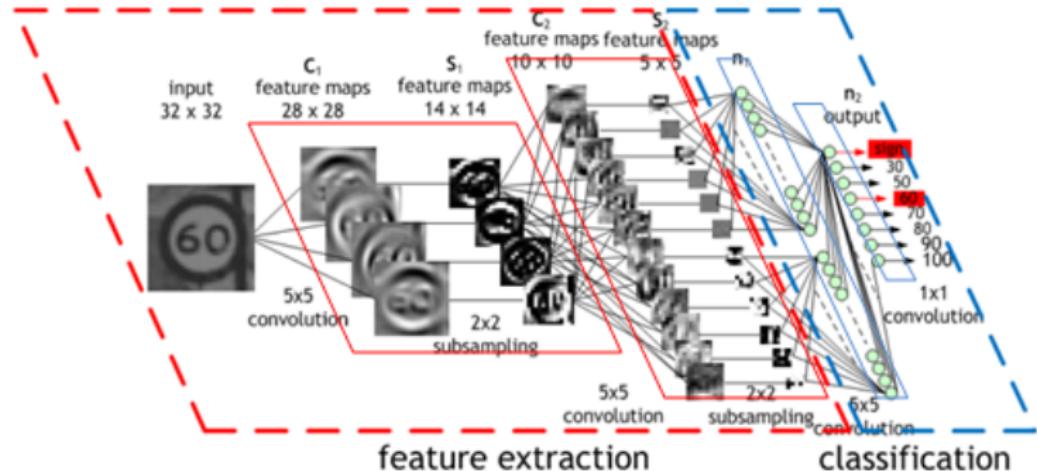
Modelo de saliência visual

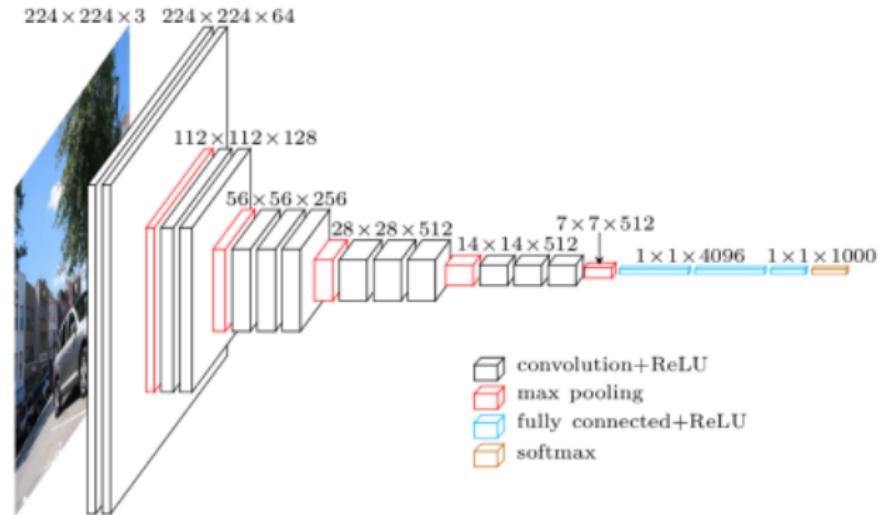
Problema:

- É difícil captar todas as nuances envolvidas na saliência em imagens



Deep Learning





*É possível um modelo de saliência visual que seja **efetivo** e
mais **eficiente**?*

Hipótese: items relevantes

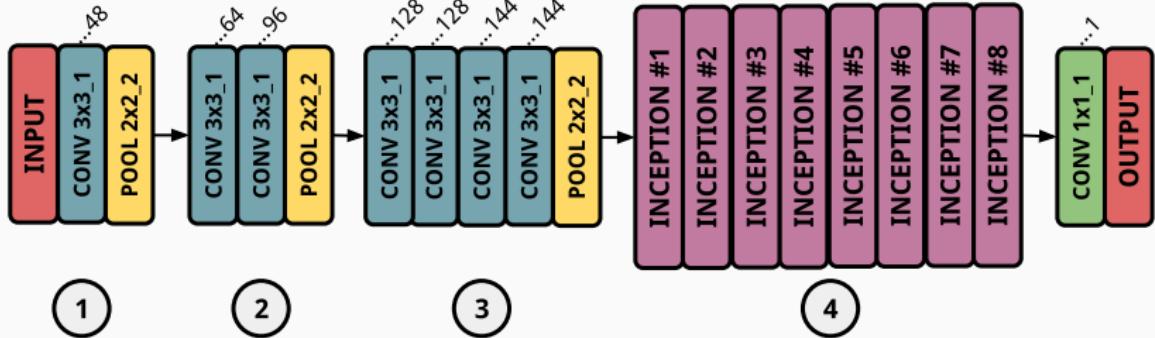
- Arquitetura de rede neural **específica** para saliência visual

Hipótese: items relevantes

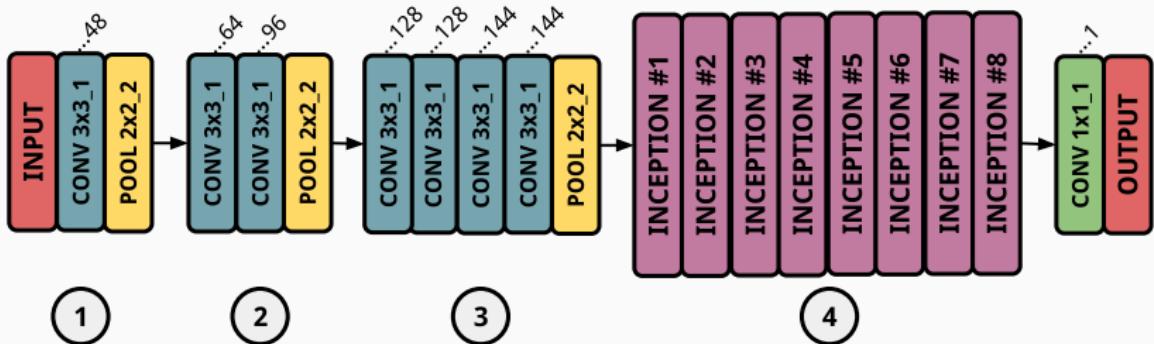
- Arquitetura de rede neural **específica** para saliência visual
- Pré-processamento de dados adequado ao contexto de saliência visual

Modelo Proposto

Arquitetura da rede

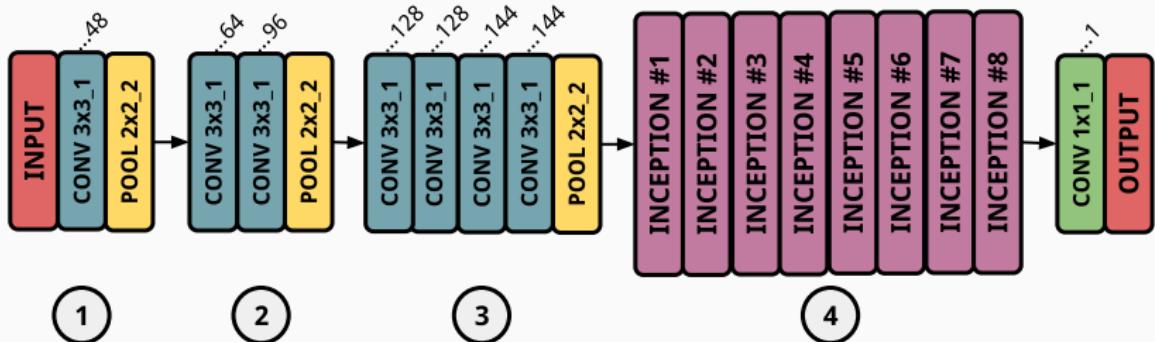


Arquitetura da rede



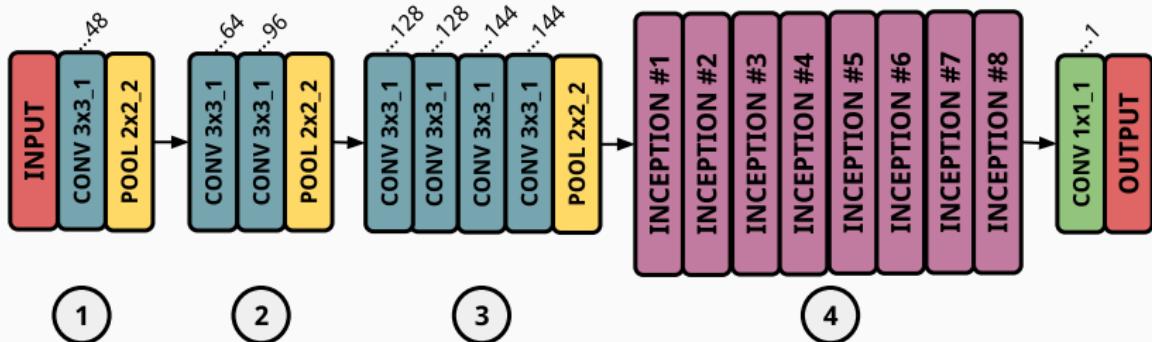
- Entrada: $(H \times W \times 3)$

Arquitetura da rede



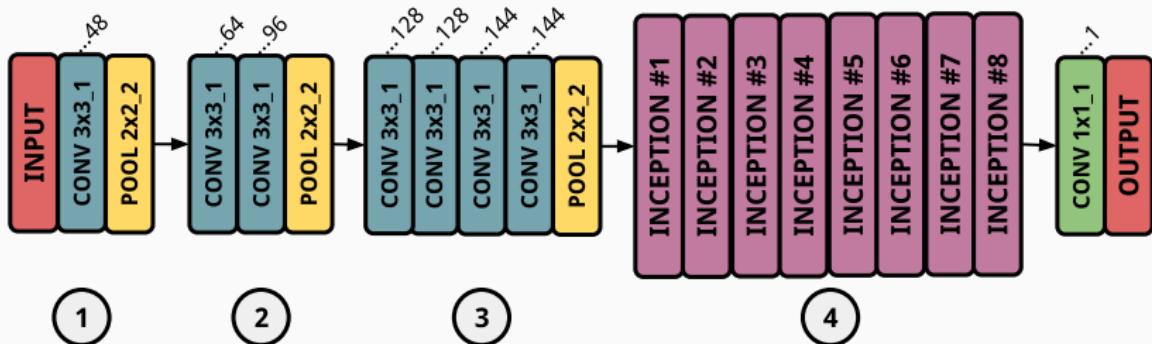
- Entrada: $(H \times W \times 3)$
- 1. Convolução/ReLU + MaxPool $(H \times W \times 3)$

Arquitetura da rede



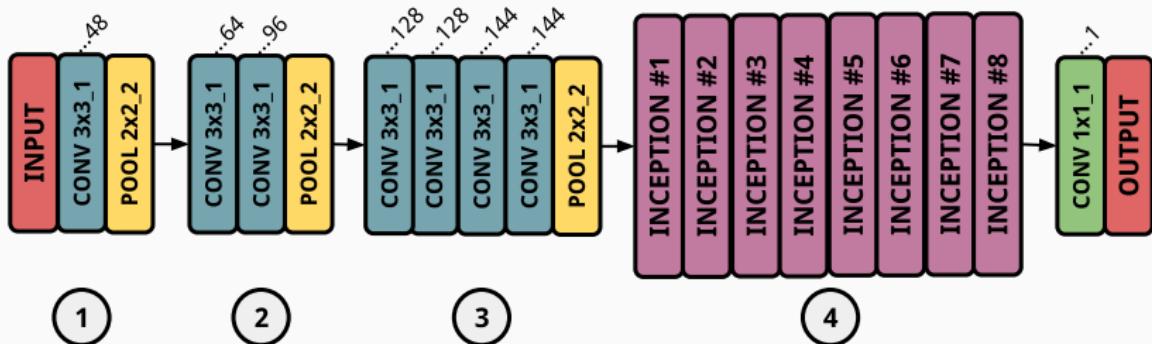
- Entrada: $(H \times W \times 3)$
- 1. Convolução/ReLU + MaxPool $(H \times W \times 3)$
- 2. 2 Convoluções/ReLU + MaxPool $(H/2 \times W/2 \times 48)$

Arquitetura da rede



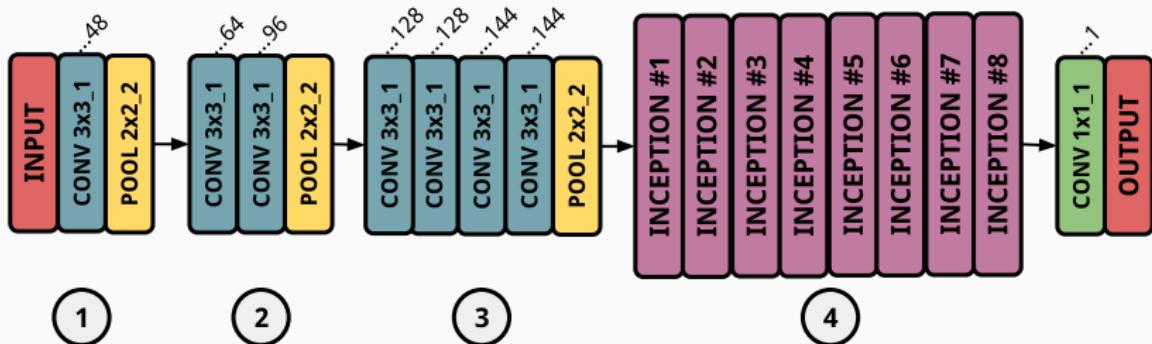
- Entrada: $(H \times W \times 3)$
- 1. Convolução/ReLU + MaxPool $(H \times W \times 3)$
- 2. 2 Convoluções/ReLU + MaxPool $(H/2 \times W/2 \times 48)$
- 3. 4 Convoluções/ReLU + MaxPool $(H/4 \times W/4 \times 96)$

Arquitetura da rede



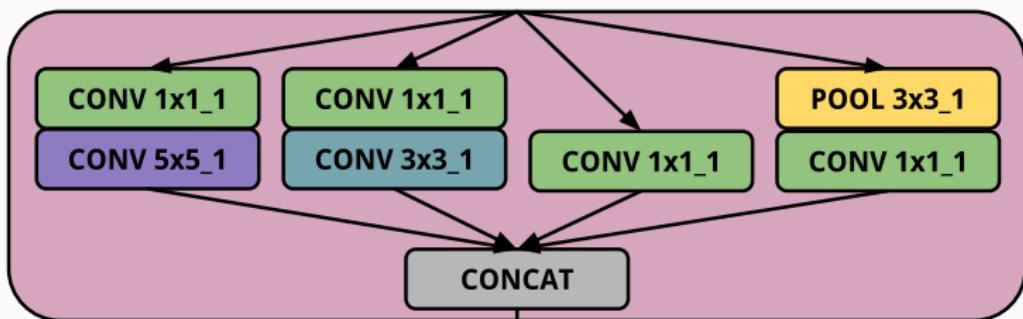
- Entrada: $(H \times W \times 3)$
- 1. Convolução/ReLU + MaxPool $(H \times W \times 3)$
- 2. 2 Convoluções/ReLU + MaxPool $(H/2 \times W/2 \times 48)$
- 3. 4 Convoluções/ReLU + MaxPool $(H/4 \times W/4 \times 96)$
- 4. 8 *inception layers* $(H/8 \times W/8 \times 144)$

Arquitetura da rede



- Entrada: $(H \times W \times 3)$
- 1. Convolução/ReLU + MaxPool $(H \times W \times 3)$
- 2. 2 Convoluções/ReLU + MaxPool $(H/2 \times W/2 \times 48)$
- 3. 4 Convoluções/ReLU + MaxPool $(H/4 \times W/4 \times 96)$
- 4. 8 *inception layers* $(H/8 \times W/8 \times 144)$
- Saída $(H/8 \times W/8 \times 1)$

Inception



Pré-processamento de dados

- Espaço de cor **LAB** ao invés de RGB

Pré-processamento de dados

- Espaço de cor **LAB** ao invés de RGB
- Normalização de canais **por imagem** ao invés de pelo *dataset*

Treinamento: Datasets

- Salicon: 15000 imagens

Treinamento: Datasets

- Salicon: 15000 imagens
- Judd: 1003 imagens *dataset*

Treinamento: Datasets

- Salicon: 15000 imagens
- Judd: 1003 imagens *dataset*
- Aumento de dados por espelhamento de imagem

Treinamento: Função objetivo

$$\min CC(P, T) = \frac{cov(P, T)}{\sigma(P)\sigma(T)}$$

Treinamento: Função objetivo

$$\min CC(P, T) = \frac{\text{cov}(P, T)}{\sigma(P)\sigma(T)}$$

- Penalização equilibrada de falsos positivos/negativos

Treinamento: Função objetivo

$$\min CC(P, T) = \frac{\text{cov}(P, T)}{\sigma(P)\sigma(T)}$$

- Penalização equilibrada de falsos positivos/negativos
- Não força a rede a produzir valores em uma certa escala

Treinamento: Etapa 1

- 30000 imagens advindas do Salicon

Treinamento: Etapa 1

- 30000 imagens advindas do Salicon
- $SGD + Nesterov\ Momentum$ de 0.9

Treinamento: Etapa 1

- 30000 imagens advindas do Salicon
- *SGD + Nesterov Momentum* de 0.9
- *Learning Rate*: 0.009 por 5 epochs, 0.001 por 3 epochs

Treinamento: Etapa 1

- 30000 imagens advindas do Salicon
- *SGD + Nesterov Momentum* de 0.9
- *Learning Rate*: 0.009 por 5 epochs, 0.001 por 3 epochs
- *Batch size*: 10

Treinamento: Etapa 2

- 2006 imagens advindas do Judd

Treinamento: Etapa 2

- 2006 imagens advindas do Judd
- $SGD + Nesterov\ Momentum$ de 0.9

Treinamento: Etapa 2

- 2006 imagens advindas do Judd
- *SGD + Nesterov Momentum* de 0.9
- *Learning Rate*: 5×10^{-5} por 2 epochs

Treinamento: Etapa 2

- 2006 imagens advindas do Judd
- *SGD + Nesterov Momentum* de 0.9
- *Learning Rate*: 5×10^{-5} por 2 epochs
- Regularização L2 de 3×10^{-5}

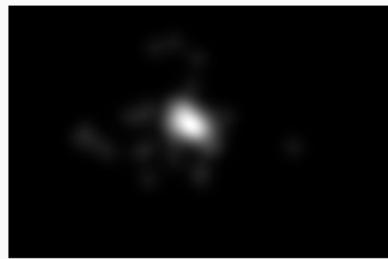
Treinamento: Etapa 2

- 2006 imagens advindas do Judd
- *SGD + Nesterov Momentum* de 0.9
- *Learning Rate*: 5×10^{-5} por 2 epochs
- Regularização L2 de 3×10^{-5}
- *Batch size*: 2

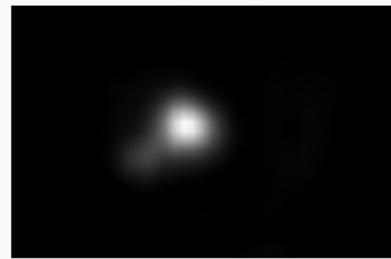
Resultados



Humanos

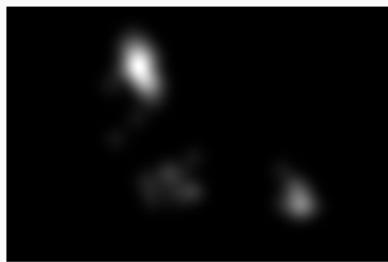


Modelo Proposto

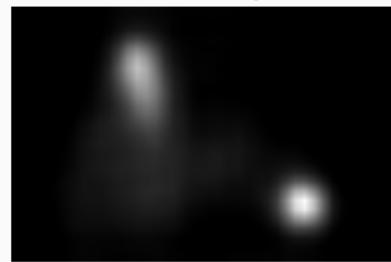




Humanos

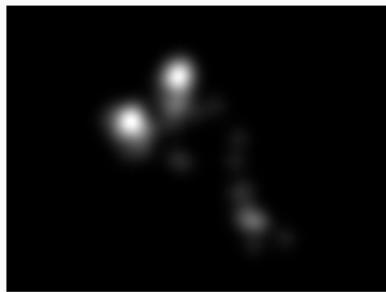


Modelo Proposto

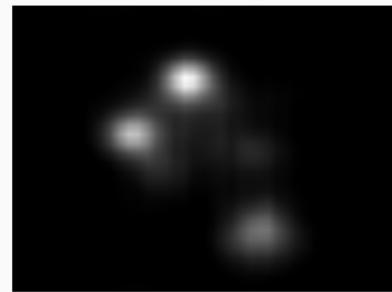




Humanos



Modelo Proposto



MIT300 Benchmark

- 300 imagens não publicadas

MIT300 Benchmark

- 300 imagens não publicadas
- Submete-se modelo e os resultados são publicados

MIT300 Benchmark: Resultados

Modelo	N. parâmetros	AUC-Judd ↑	CC ↑	Sim ↑
<i>DeepFix</i>	≈16.7 M	0.87	0.78	0.67
<i>Salicon</i>	≈14.7 M	0.87	0.74	0.60
Modelo Proposto	3.72 M	0.85	0.71	0.62
<i>ML-Net</i>	≈15.4 M	0.85	0.69	0.60
<i>SalNet</i>	25.8 M	0.83	0.57	0.52

Conclusões

- É possível obter redes convolucionais mais eficientes para detecção de saliência visual

Conclusões

- É possível obter redes convolucionais mais eficientes para detecção de saliência visual
- Nosso modelo tem desempenho comparável ao estado da arte tendo cerca de **1/4** do número de parâmetros

Conclusões

- É possível obter redes convolucionais mais eficientes para detecção de saliência visual
- Nosso modelo tem desempenho comparável ao estado da arte tendo cerca de **1/4** do número de parâmetros
- Arquitetura e pré-processamento de dados **específicos para o contexto de saliência visual** mostraram-se importantes

Obrigado!

Referências

- Treisman, Gelade. A Feature-Integration theory of Attention. *Cognit Psychol*, 1980.
- Zoya, Judd, 2015. MIT Saliency Benchmark.
- Kruthiventi et al, 2015. DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations. *arXiv preprint arXiv:1609.01064*. 2016.
- Jiang. Salicon: saliency in context. *CVPR* 2015
- Simone Frintrop. VOCUS: a visual attention system for object detection and goal-directed search. 2005.