

# Finding Waldo

Rešavanje problema binarne  
klasifikacije pomoću različitih modela  
mašinskog učenja

Aleksandar Anžel 1025/2018  
Matematički fakultet  
Univerzitet u Beogradu

# Rešen problem



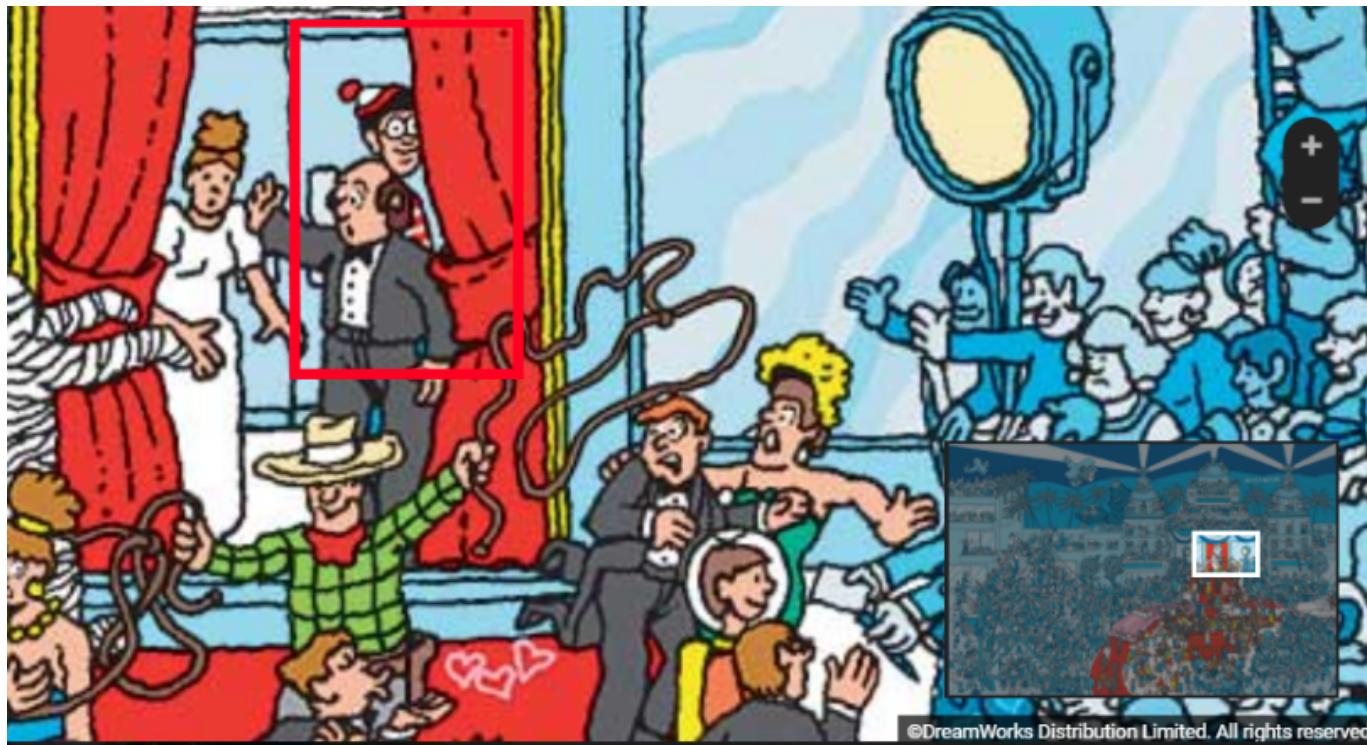
©DreamWorks Distribution Limited. All rights reserved.

# Rešen problem

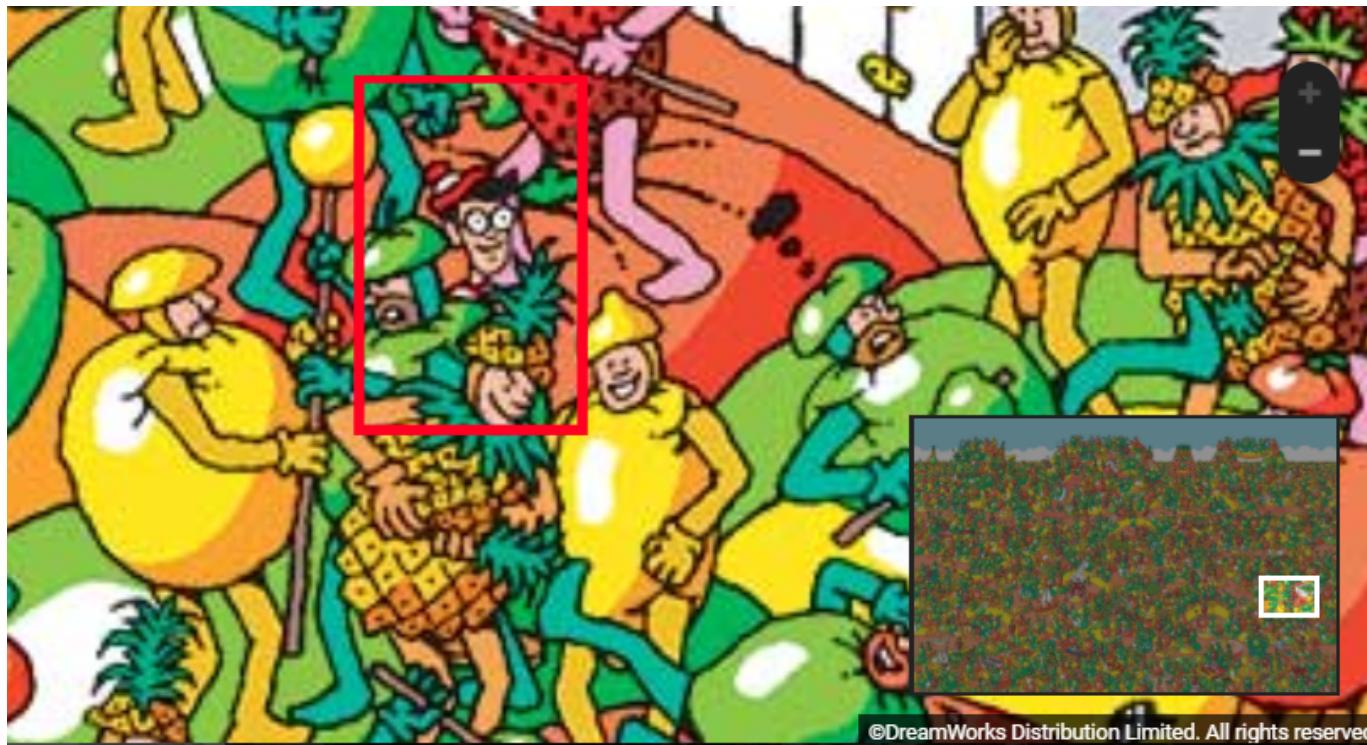


©DreamWorks Distribution Limited. All rights reserved.

# Rešen problem



# Rešen problem



©DreamWorks Distribution Limited. All rights reserved.

# Gde je Valdo?



# Skup podataka

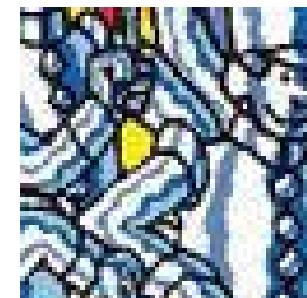
- Napravljen od 19 slika velike rezolucije
- Slike razbijene na skupove različitih formata:
  - 256 x 256 piksela (317 slika)
  - 128 x 128 piksela (1344 slika)
  - 64 x 64 piksela (5376 slika) - korišćen u projektu
- [Link ka GitHub direktorijumu](#)

# Prikaz skupa podataka

- Valdo:

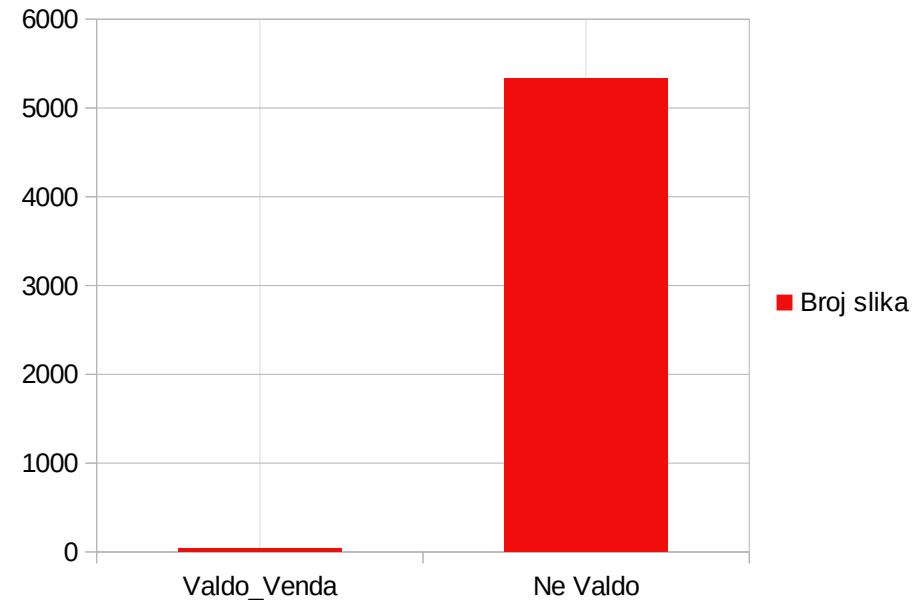


- Ne Valdo:



# Balansiranost skupa podataka

- Broj slika na kojima su Valdo ili Venda: 39
- Broj slika na kojima se oni ne nalaze:  
 $5376 - 39 = 5337$



**Rešavanje problema pomoću:**  
Logističke regresije  
Metoda potpornih vektora  
K najbližih suseda

# Povećanje skupa podataka

- Balansiranje skupa podataka izmenom postojećih slika
- `from keras_preprocessing.image import ImageDataGenerator`
- `flow_from_directory`
- Od 39 do ~5300 slika na kojima su Valdo ili Venda

# Izmenjeni podaci



# Priprema podataka

- Ravnanje slika (eng. *Flatten*)
- Pakovanje u DataFrame (*pandas* biblioteka) za potrebe ponovnog korišćenja
- Podele na skupove za trening, validaciju i testiranje
- Centriranje vrednosti piksela

# Priprema podataka

- Primena analize glavnih komponenti
  - Čuvanje 3000 glavnih komponenti
  - Ukupna sačuvana varijansa: 73.80170621466442 %
  - Vreme treniranja je smanjeno od 24+h na ~8h

# Treniranje

- Konfiguracije pomoću **GridSearchCV** (*sklearn* biblioteka):
  - K najbližih suseda:
    - Broj suseda: [3, 4, 5, 6, 7, 8, 9, 10]
  - Metod potpornih vektora:
    - C: [ $10^{**i}$  for i in range(-3, 3)]
    - Gama: [ $10^{**i}$  for i in range(-3, 3)]

# Rezultati

- Tačnosti (eng. Accuracy) nad test skupom:
  - Logistička regresija: 0.49199417758369723
  - K najbližih suseda: 0.5027656477438137
  - Metod potpornih vektora: 0.5129548762736535
- Zaključak:
  - Nema smisla testirati lociranje Valda nad novom slikom
  - Modeli nisu pogodni za rešavanje ovakvog problema

**Rešavanje problema pomoću:  
Konvolutivnih neuronskih mreža**

# Povećanje skupa podataka

- Treniranje mreže nad istim skupom daje slične rezultate
- Skup podataka je premali
- Povećanje uz izmenu nije dovoljno
- Drugi pristup

# Generisanje novih slika



# Transparentne slike Valda



# Povećanje skupa podataka

- Balansiranje skupa podataka lepljenjem transparentnih slika preko slika gde nema Valda
- `from keras_preprocessing.image import  
ImageDataGenerator`
- `flow_from_directory`
- Od 5300 slika na kojima je Valdo i na kojima nije do ~150000

# Korišćen model

Layer (type)	Output Shape	Param #
=====		
conv2d_1 (Conv2D)	(None, 64, 64, 32)	416
activation_1 (Activation)	(None, 64, 64, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout_1 (Dropout)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 31, 31, 64)	8256
activation_2 (Activation)	(None, 31, 31, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 15, 15, 64)	0
=====		

Layer (type)	Output Shape	Param #
=====		
flatten_1 (Flatten)	(None, 14400)	0
dense_1 (Dense)	(None, 64)	921664
activation_3 (Activation)	(None, 64)	0
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65
activation_4 (Activation)	(None, 1)	0
=====		

Total params: 930,401

Trainable params: 930,401

Non-trainable params: 0

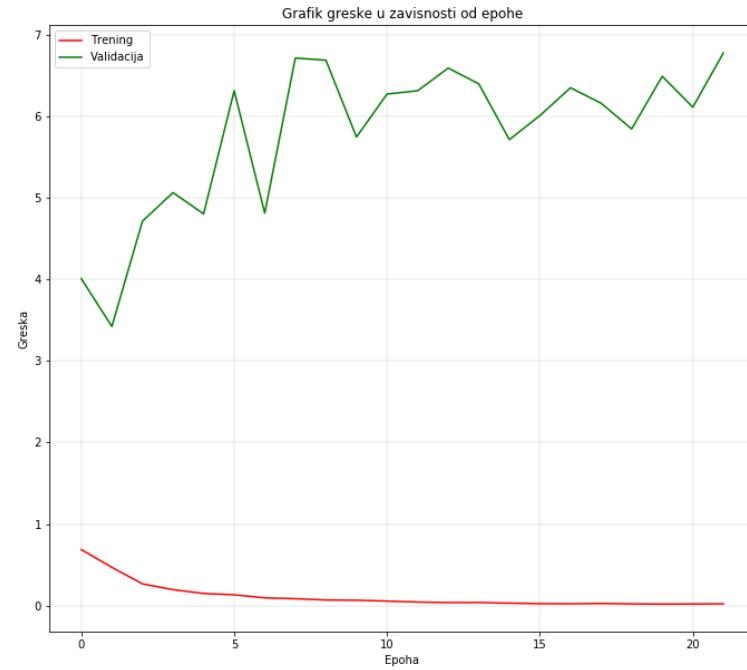
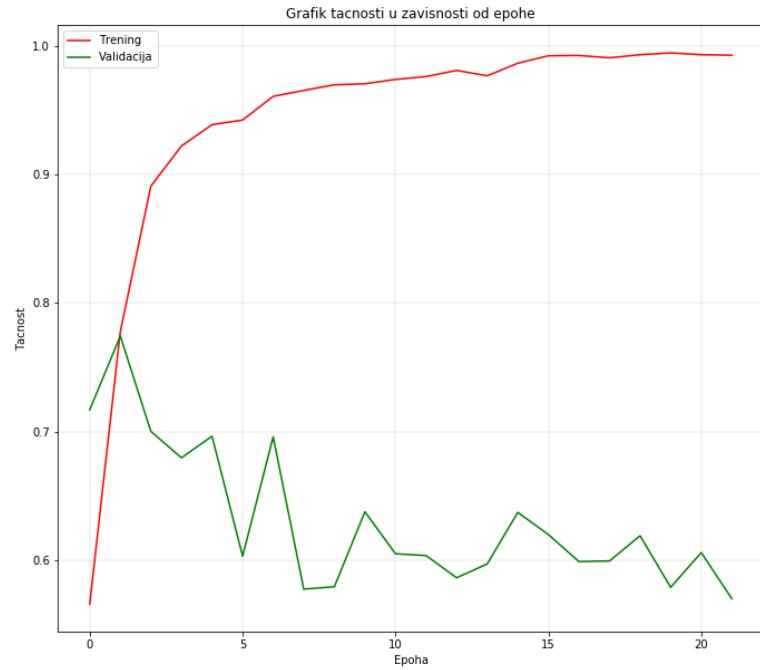
# Treniranje

- Regularizacija:
  - Izostavljanjem ćelija (eng. *Dropout*)
    - Rate: 0.3
  - Ranim zaustavljanjem (eng. *Early Stopping*)
    - Patience: 20
- Optimizacija:
  - Adam ( $\text{lr} = 0.00075$ )
- Greška:
  - Binarna krosentropija
- Metrika:
  - Tačnost
- Broj epoha:
  - 1000

# Rezultati

- Tačnost:
  - Nad trening skupom: 0.9994005994005994
  - Nad test skupom: 0.9960261141072949
- Greška:
  - Nad trening skupom: 0.01121072506375767
  - Nad test skupom: 0.02126941974905301

# Rezultati



# Zaključak

- Težak problem
- Skup podataka je premali
- Skup podataka sadrži i slike lošijeg kvaliteta
- Lik Valda na slikama nije centriran, velike varijacije položaja
- Preprilagođavanje
- Bolji rezultati uz označavanje svake od slike

# Ideja lociranja Valda na slici (pod uslovom da je model dovoljno dobar)

- 1) Novu sliku izdeliti na delove dimenzije 64x64
- 2) Ime svakog dela sadrži koordinate gornjeg levog ugla tog dela u polaznoj slici
- 3) Odraditi predikciju modela nad skupom delova
- 4) Za one delove za koje model daje odgovor  $>0.5$  izvući njihove koordinate iz imena
- 5) Obeležiti pravougaonikom lokaciju tog dela u polaznoj slici

# Drugačiji pristupi

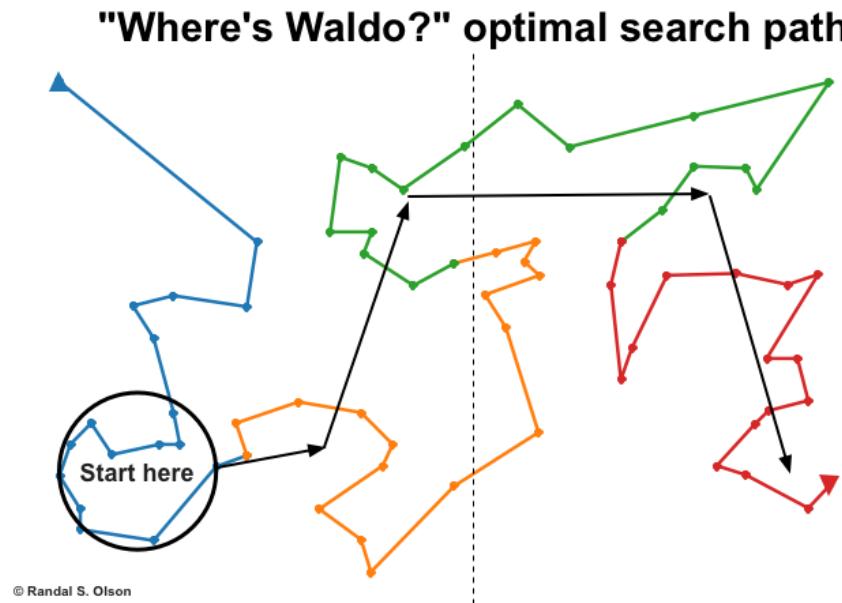
- Transfer učenja
- Traženje optimalne putanje
- Korišćenje složenijeg modela
- ...

# Transfer učenja

- Model: ***R-CNN with Inception v2*** treniran na **COCO** skupu podataka
- Tensorflow Object Detection API

[LINK](#)

# Traženje optimalne putanje

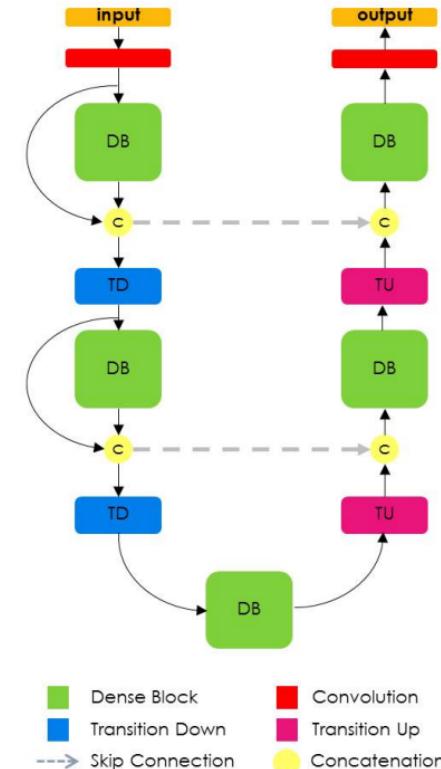


- 1) VIDEO LINK
- 2) LINK

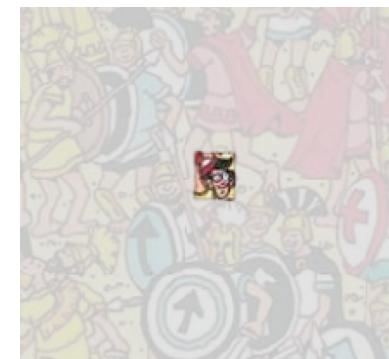
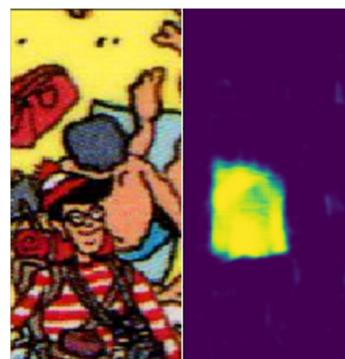
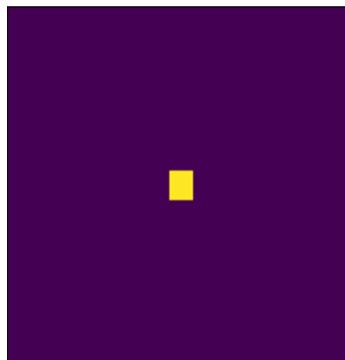
# Korišćenje složenijeg modela

- TIRAMISU ([LINK](#))

The One Hundred Layers Tiramisu:  
Fully Convolutional DenseNets for  
Semantic Segmentation



# Korišćenje složenijeg modela



# Hvala na pažnji!

