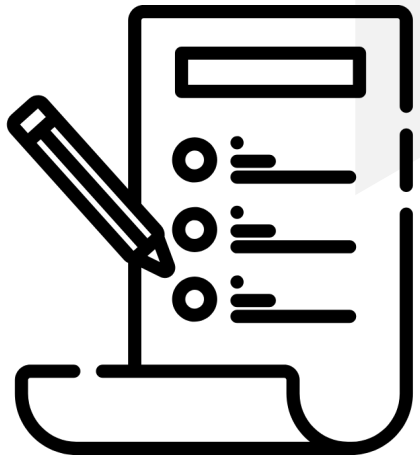
Recognition System

Digital Signal & Image Management
Final Project 2018/19

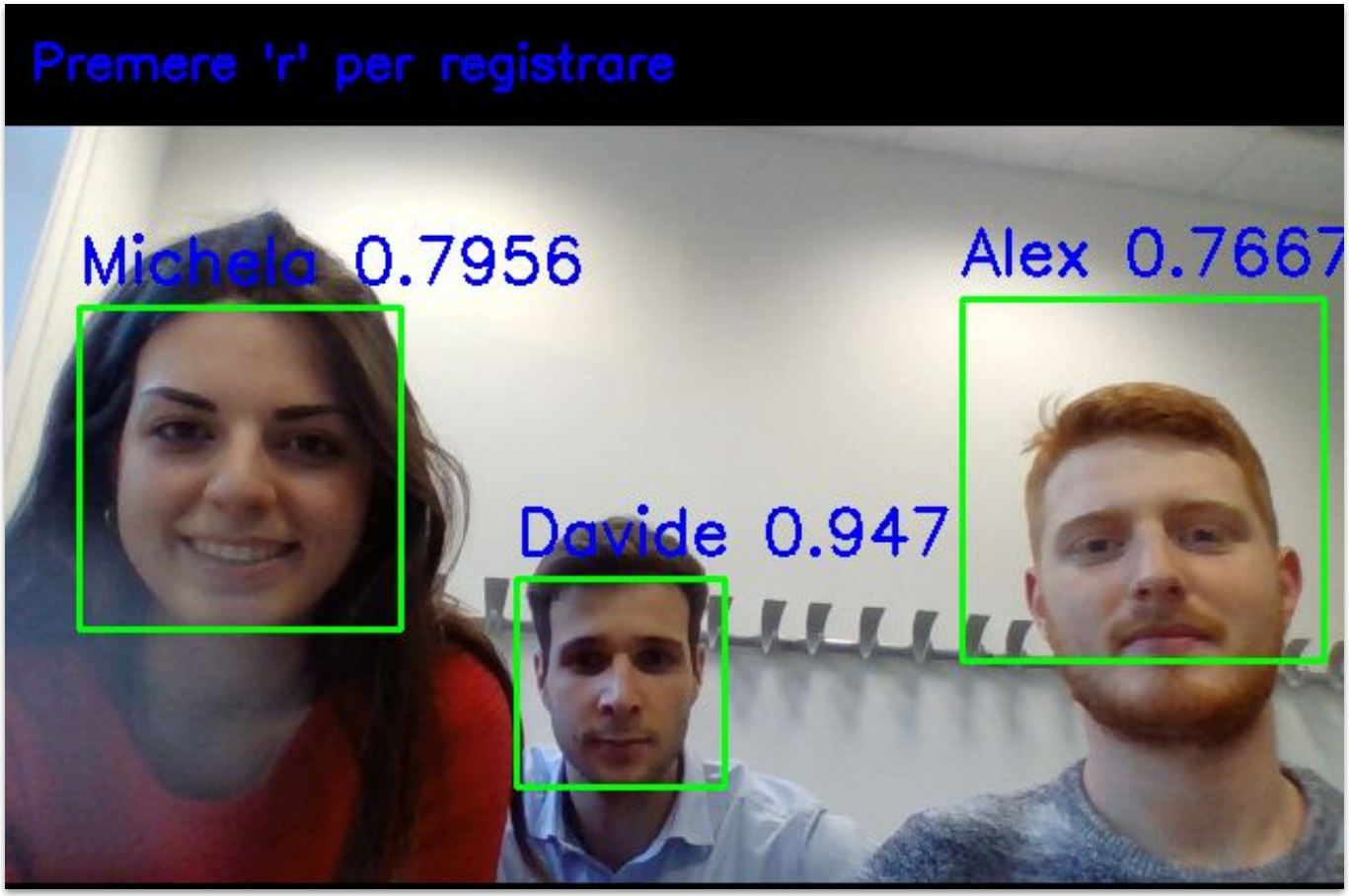
Agenda

1. **Il team**
2. **Obiettivi**
3. **L’idea di implementazione**
4. **Approccio metodologico**
    - Overview del processo
    - Step 1: **Data acquisition**
    - Step 2: **Audionet**
    - Step 3: **Recognet**
    - Step 4: **Retrieval**
5. **Demo Live**
6. **Improvements**



Il Team

Presentazione del team di lavoro



Michela Sessi
Mat. 777760

Davide Pecchia
Mat. 793290

Alex Ceccotti
Mat. 790497

Obiettivi

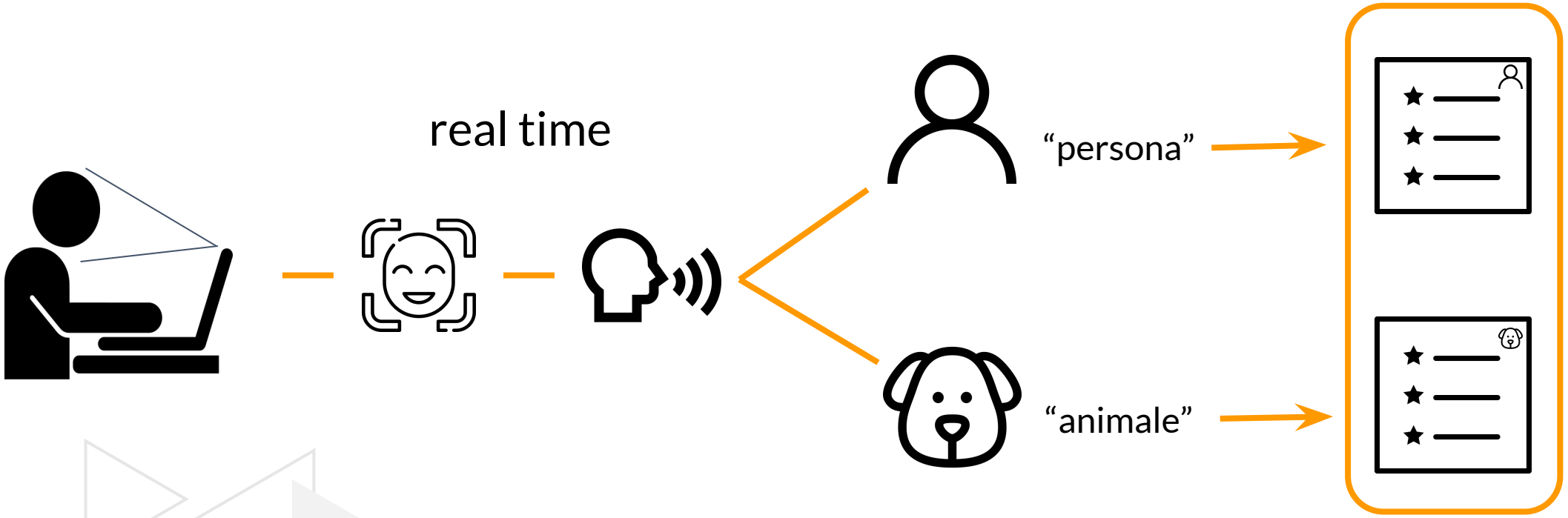
Task 1: riconoscere a partire da un file audio, l'identità di chi sta parlando. Oltre al riconoscimento della persona, dovrà essere riconosciuta anche la parola pronunciata.

Task 2: riconoscere l'identità di una persona a partire da una foto del volto. Il riconoscimento dovrà avvenire anche live da una webcam.

Task 3: trovare all'interno di un dataset fornito dal docente i 10 volti che somigliano di più a quello del punto precedente. I 10 risultati restituiti devono essere ordinati per similarità decrescente. In aggiunta, in base alla scelta dell'utente, saranno forniti i 10 cani a cui si assomiglia di più.

L’idea

Struttura ideale del progetto

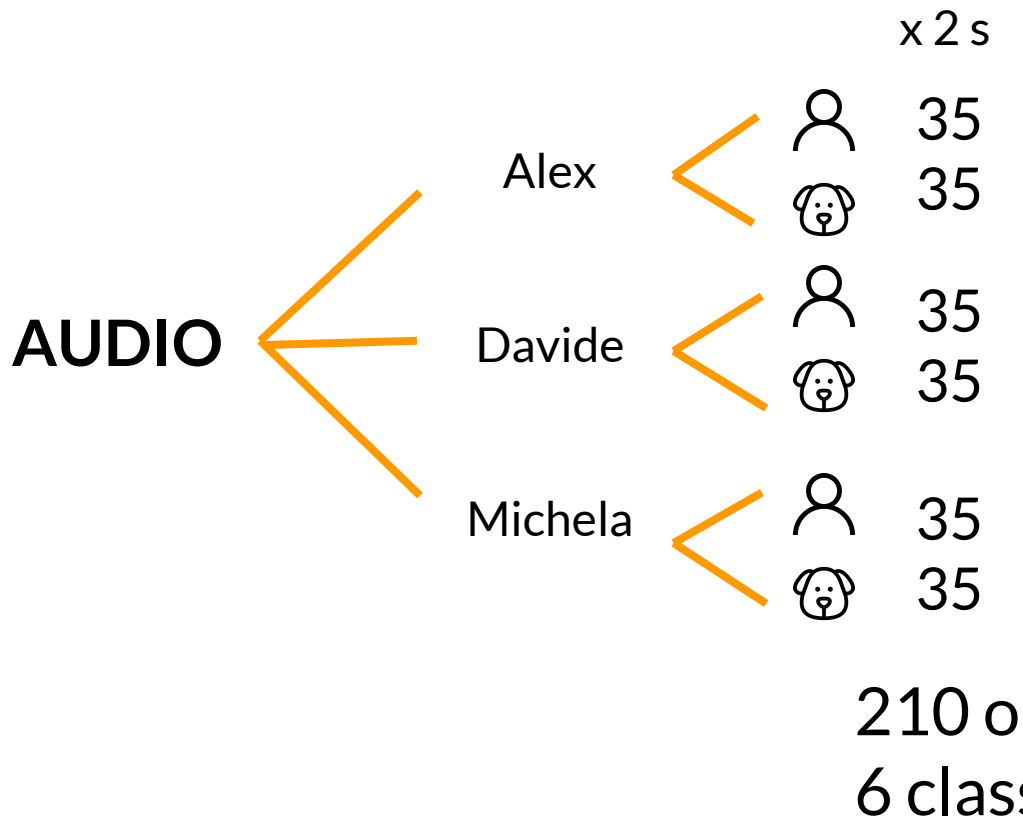


Overview del processo

Acquisition

Acquisizione automatica di audio e video

1 Data Acquisition



210 obs
6 classes

Gestione del rumore

Acquisition

Acquisizione automatica di audio e video

1 Data Acquisition

VIDEO

	1	2
Alex	500	500
Davide	500	500
Michela	500	500

3000 obs

Face detector

Audionet

Neural Network per la classificazione degli audio

2 Implementazione Audionet

1 Manipolazione audio

audio di 2 sec
rate per sec: 44100

coefficienti **MFCC**

per ogni audio:
173 x 20
time steps x features

2 Partizione del Dataset

80% Training set

20% Test set

3 Costruzione Rete Neurale

Recurrent Neural Network

40 celle GRU

4 Addestramento e valutazione metriche

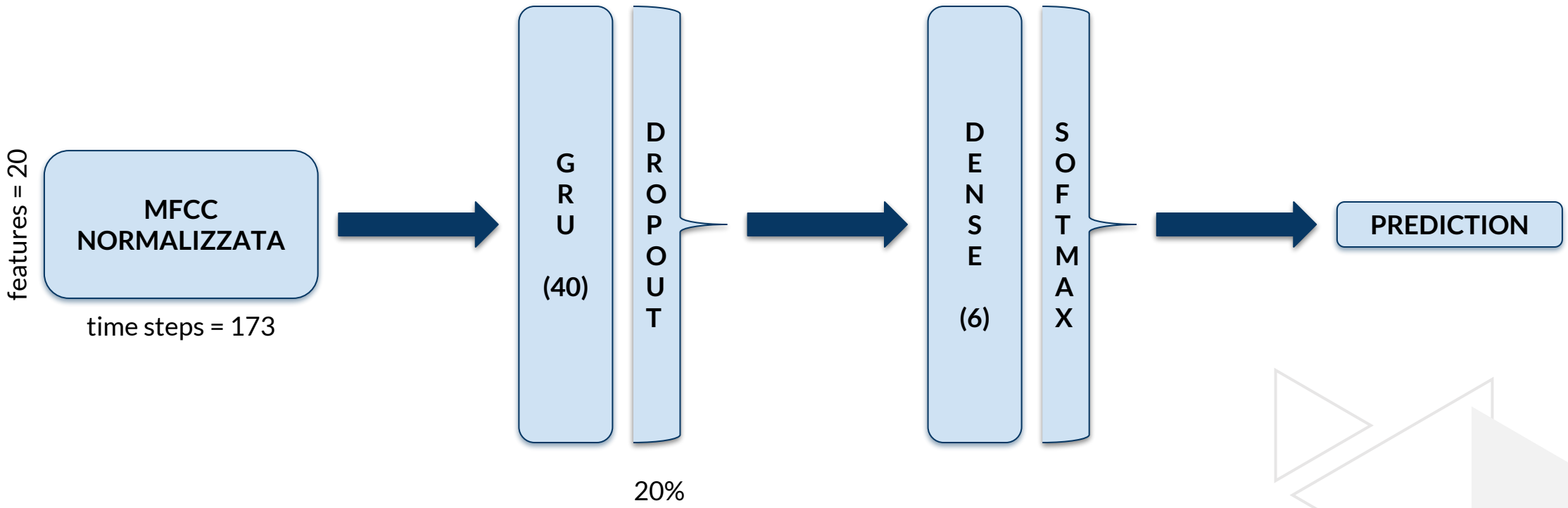
- **batch size** = 16
- **300 epoche** di training
- **10% Validation set**

metrica di valutazione finale: **Accuracy**.

Audionet

Neural Network per la classificazione degli audio

2 Implementazione Audionet



Optimizer RMSprop with:
- learning rate = 0.001
- decay = 0.001
- rho = 0.9

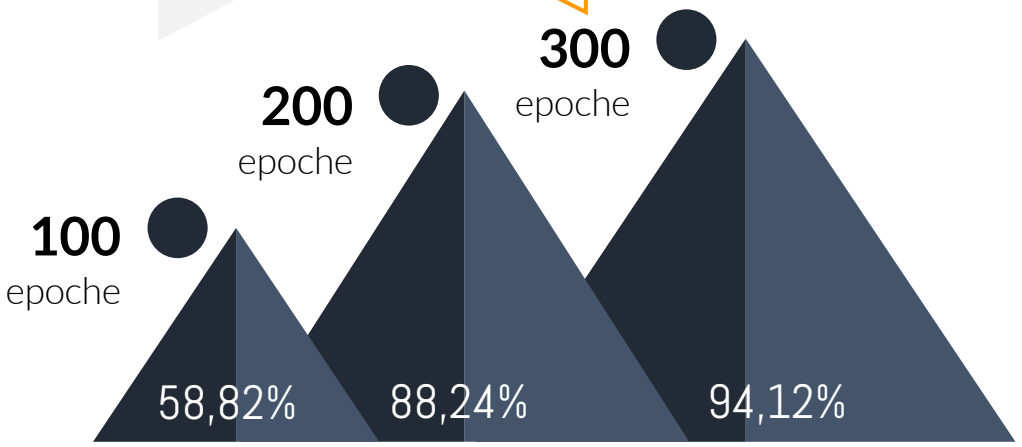
https://arxiv.org/pdf/1811.07453v1.pdf

Audionet

Neural Network per la classificazione degli audio

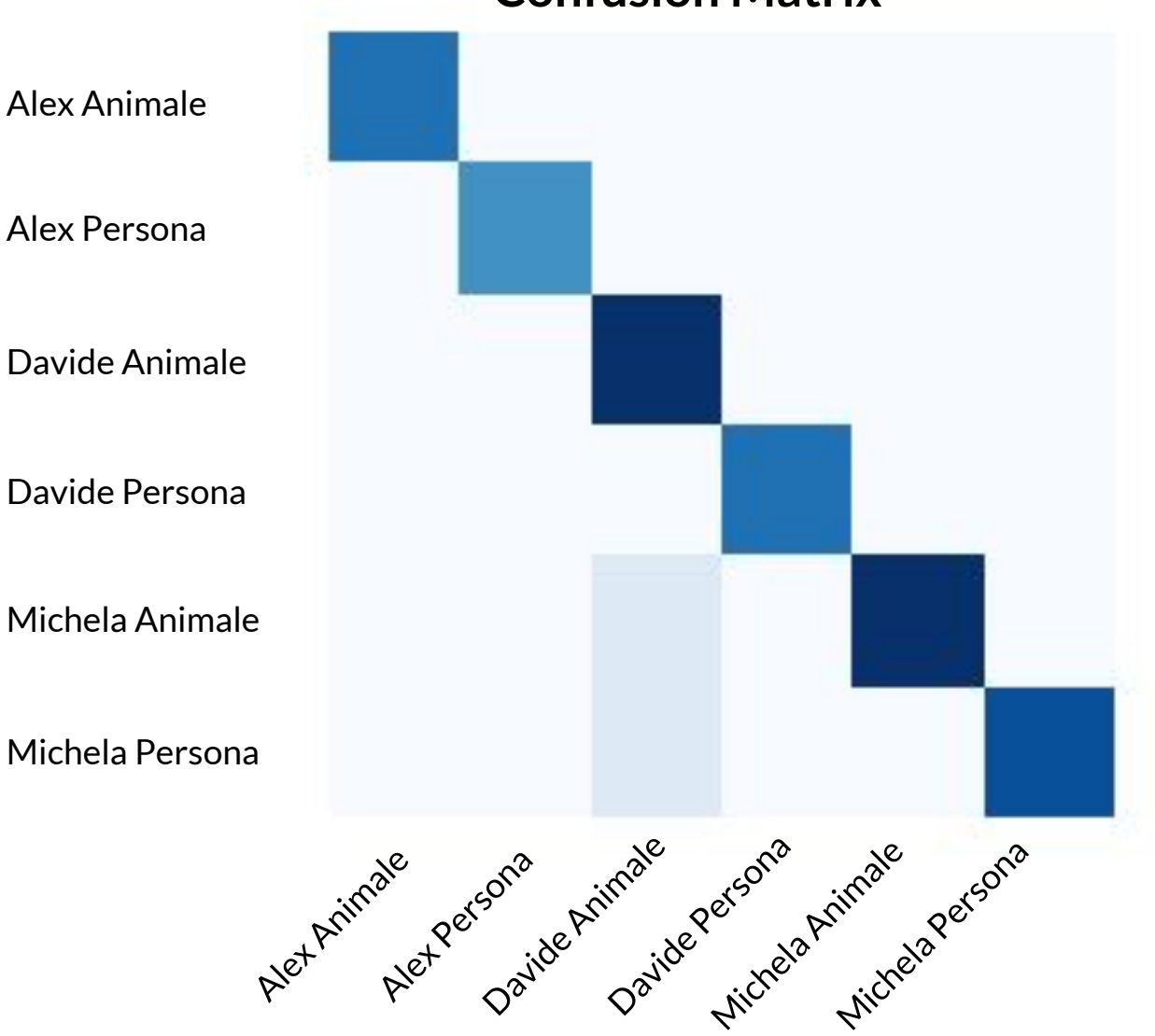
2 Implementazione Audionet

Test accuracy: 95,24%



Validation accuracy

Confusion Matrix



Recognet

Neural Network per la classificazione dei volti

1. Data Augmentation
2. Partizione del Dataset
3. Costruzione Rete Neurale
4. Addestramento e valutazione metriche

Recognet

Neural Network per la classificazione dei volti

3 Implementazione Recognet

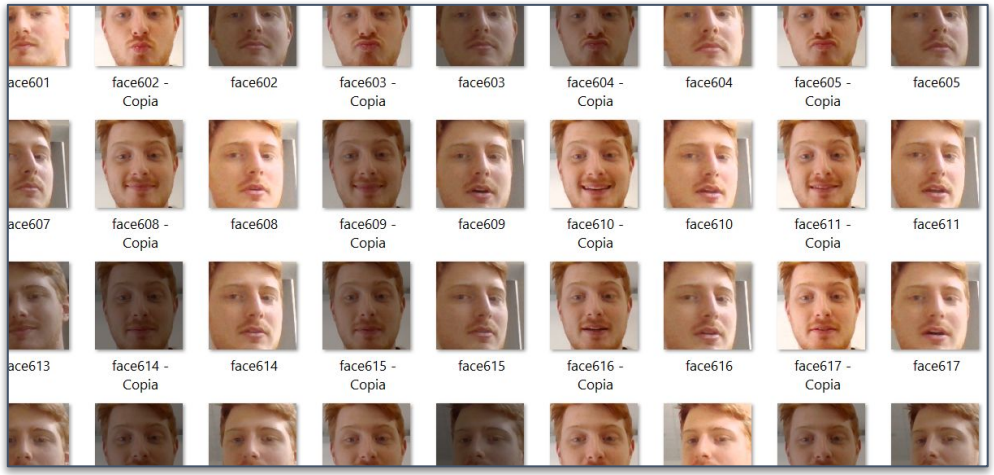
1

Data Augmentation

- Horizontal flip
- Vertical flip
- Translation
- Rotation
- Adding random noise
- Pre-process: image_contrast_adjustment

→ **Brightness**

https://medium.com/@vivek.yadav/improved-performance-of-deep-learning-neural-network-models-on-traffic-sign-classification-using-6355346da2dc



~6000 obs, 3 classes

Recognet

Neural Network per la classificazione dei volti

2

Partizione del Dataset

~6000 obs, 3 classes

Kimage.ImageDataGenerator

resnet50.preprocess_input
Target size: 224x224
Color mode: RGB

→ Batch size: 173

75%	25%
Training set	Validation set
4498	1500

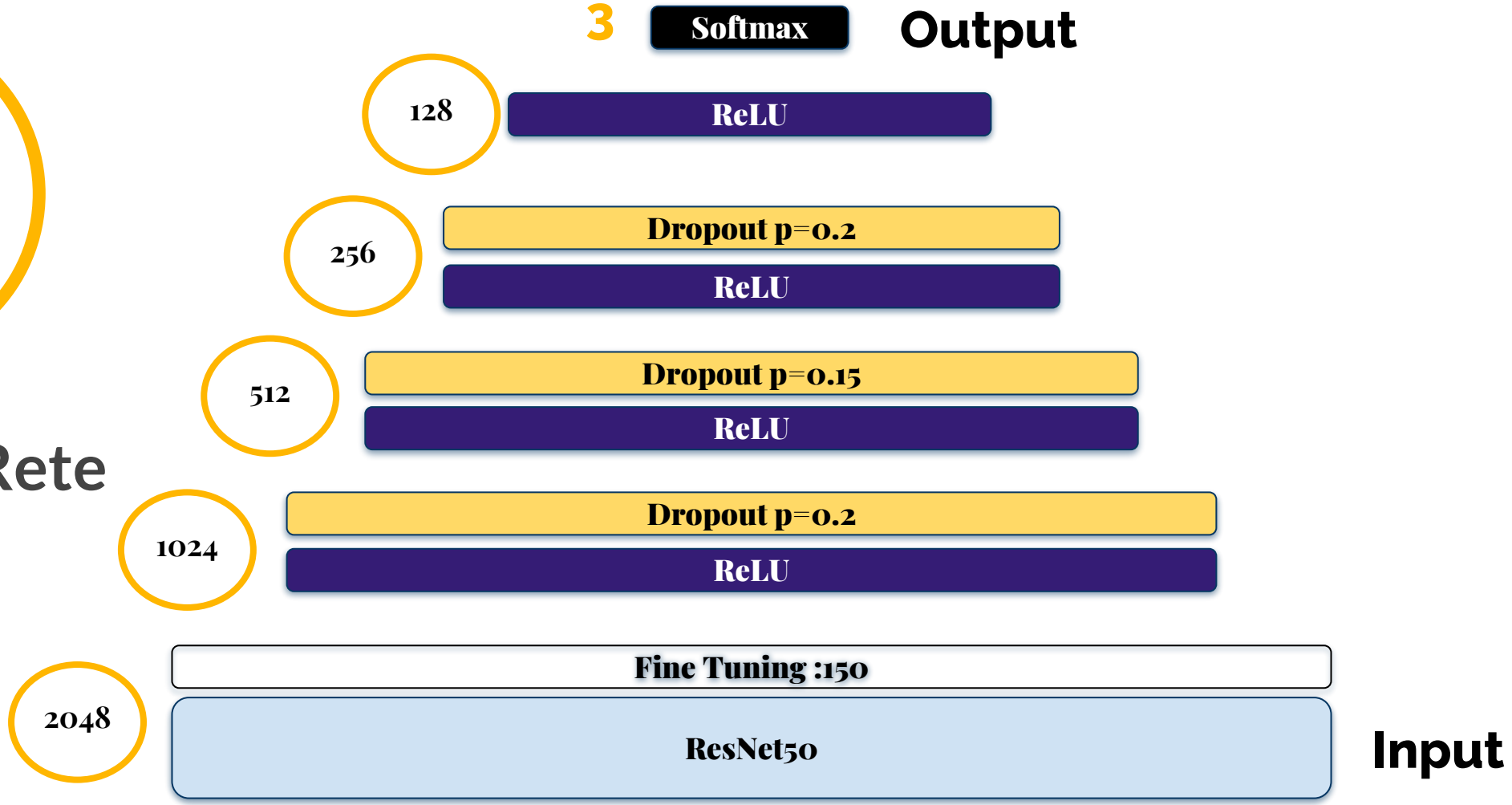
https://www.quora.com/Should-I-use-powers-of-2-when-choosing-the-size-of-a-batch-size-when-training-my-Neural-Network

Recognet

Neural Network per la classificazione dei volti

3

Costruzione Rete Neurale



Recognet

Neural Network per la classificazione dei volti

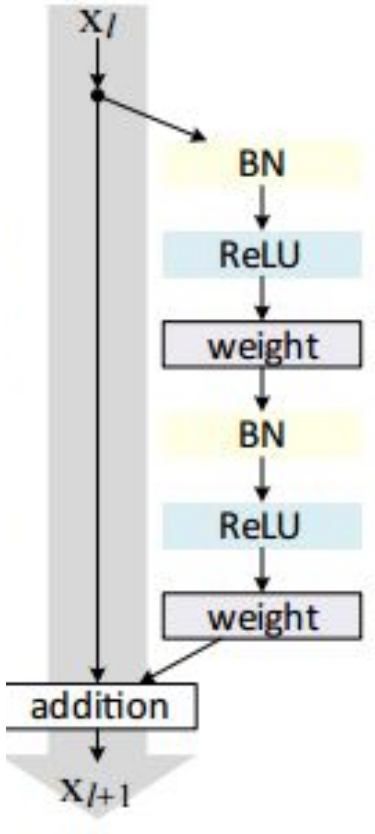
3 Implementazione Recognet

ResNet50

176 layers

Pretrain: Imagenet
Fine-Tuning dal :150

Pooling average



Layer (type)	Output Shape	Param #	Connected to
add_43 (Add)	(None, 14, 14, 1024)	0	bn4d_branch2c[0][0] activation_129[0][0]
activation_132 (Activation)	(None, 14, 14, 1024)	0	add_43[0][0]
res4e_branch2a (Conv2D)	(None, 14, 14, 256)	262400	activation_132[0][0]
bn4e_branch2a (BatchNormalizati	(None, 14, 14, 256)	1024	res4e_branch2a[0][0]
activation_133 (Activation)	(None, 14, 14, 256)	0	bn4e_branch2a[0][0]
res4e_branch2b (Conv2D)	(None, 14, 14, 256)	590080	activation_133[0][0]
bn4e_branch2b (BatchNormalizati	(None, 14, 14, 256)	1024	res4e_branch2b[0][0]
activation_134 (Activation)	(None, 14, 14, 256)	0	bn4e_branch2b[0][0]
res4e_branch2c (Conv2D)	(None, 14, 14, 1024)	263168	activation_134[0][0]
bn4e_branch2c (BatchNormalizati	(None, 14, 14, 1024)	4096	res4e_branch2c[0][0]
add_44 (Add)	(None, 14, 14, 1024)	0	bn4e_branch2c[0][0] activation_132[0][0]

- https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624
- https://towardsdatascience.com/understanding-residual-networks-9add4b664b03
- https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8

Recognet

Neural Network per la classificazione dei volti

Callbacks:

- **ModelCheckpoint**
  val_loss, min
- **EarlyStopping**
  val_loss, patience 3

Optimizer: ADAM

- https://shaoanlu.wordpress.com/2017/05/29/sgd-all-which-one-is-the-best-optimizer-dogs-vs-cats-toy-experiment/
- http://ruder.io/optimizing-gradient-descent/

in Dense(ReLU):

- **kernel_regularizer l2(0.01)**
  0.01= how we penalize higher parameters values
- **bias_regularizer l1(0.01)**

- https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/
- https://letslearnai.com/2018/03/10/what-are-l1-and-l2-loss-functions.html

Recognet

Neural Network per la classificazione dei volti

4

Addestramento e valutazione metriche

5 Epoche
26 Steps per Epoch

Retrieval

Algoritmo per trovare immagini somiglianti

1. **Caricamento Facenet**

   https://arxiv.org/pdf/1503.03832.pdf

2. **Utilizzo di Facenet**

   example: https://sefiks.com/2018/09/03/face-recognition-with-facenet-in-keras/

3. **Stima KDTree**

Retrieval

Algoritmo per trovare immagini somiglianti

1

Caricamento Facenet

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Retrieval

Algoritmo per trovare immagini somiglianti

Utilizzo Facenet

Normalizzazione

- Dati di input vengono normalizzati (richiesto per utilizzo di facenet)

Caricamento immagini persone

- Caricamento delle immagini di persone con preprocessing che consiste in:
  - conversione dell'immagine in bianco e nero
  - trovare solo il volto all'interno dell'immagine
  - estrazione delle features che vengono salvate in vettori di dimensione 128
- Considerata solo un'immagine per personaggio

Caricamento immagini cani

- Caricamento delle immagini di cani con stesso preprocessing.
- Considerata solo un'immagine frontale per razza selezionata manualmente (altrimenti rilevazione anche del padrone del cane).

Dataset attori: http://vis-www.cs.umass.edu/lfw/

Dataset cani: http://vision.stanford.edu/aditya86/ImageNetDogs/

Retrieval

Algoritmo per trovare immagini somiglianti

3

Stima KDTree

- Utilizzo del pacchetto KDTree della libreria sklearn.neighbors per stimare le **Gaussian Kernel Density** delle immagini di persone e cani.
- Nella fase di query verranno estratti i 10 nearest neighbors (sulla kernel density) dell'immagine di input. Viene utilizzato albero per ottimizzare velocità di ricerca.


Demo Live

Improvements

Conclusioni e possibili sviluppi

- Aumentare le foto:
  → diverse webcam, diversi strumenti, diversi ambienti, diverse giornate.

- Aumentare gli audio:
  → diversi microfoni, diversi giorni, diversi toni, gestione del rumore.

- Capacità computazionali maggiori per la gestione dell'aumento del dataset

- Fine-tuning più avanzato delle reti

Grazie!

Digital Signal & Image Management
Final Project 2018/19