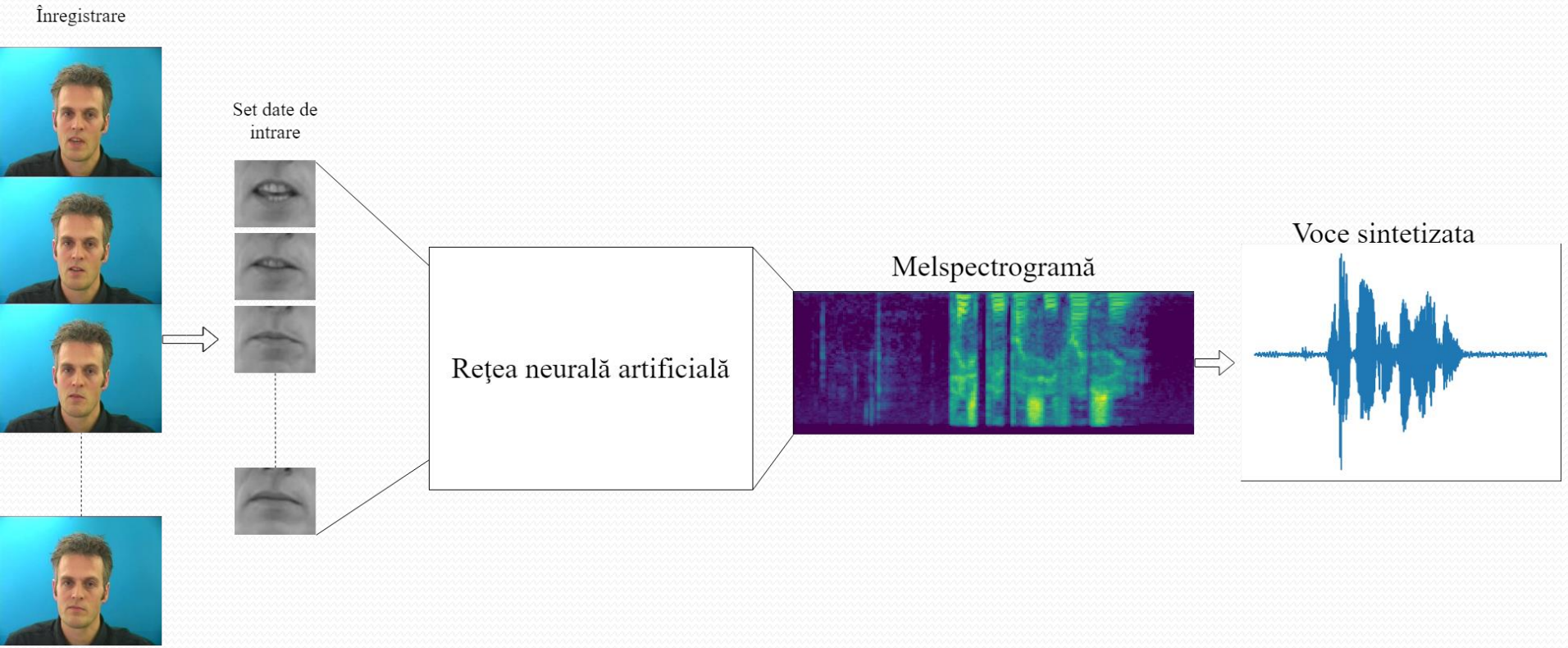
Sinteza vorbirii pornind de la mișcarea buzelor

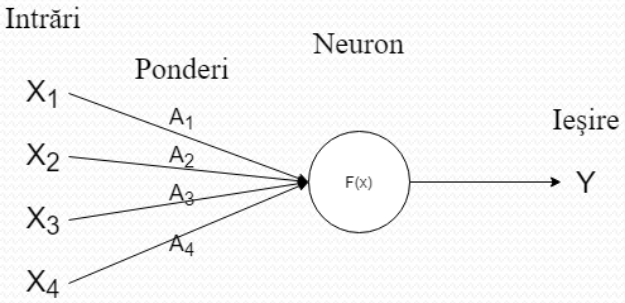
Conducători ştiinţifici
Dr. Ing. Dan ONEAȚĂ
Conf. Dr. Ing. Horia CUCU

Absolvent
Octavian PASCU

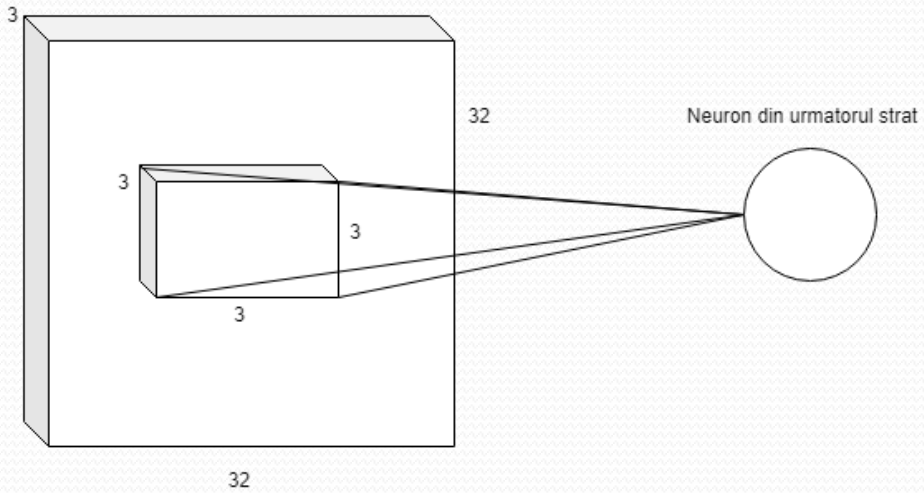
Scopul şi structura lucrării



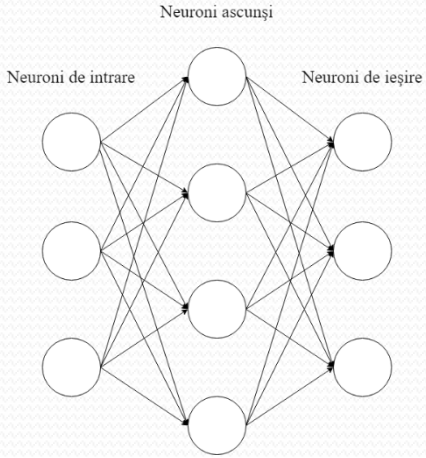
Noţiuni teoretice



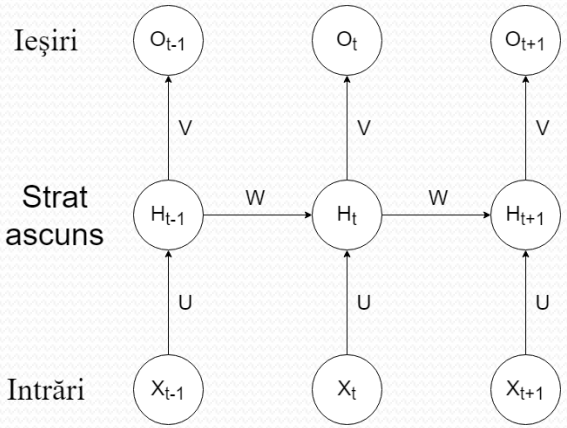
Reţea neurală convoluţională



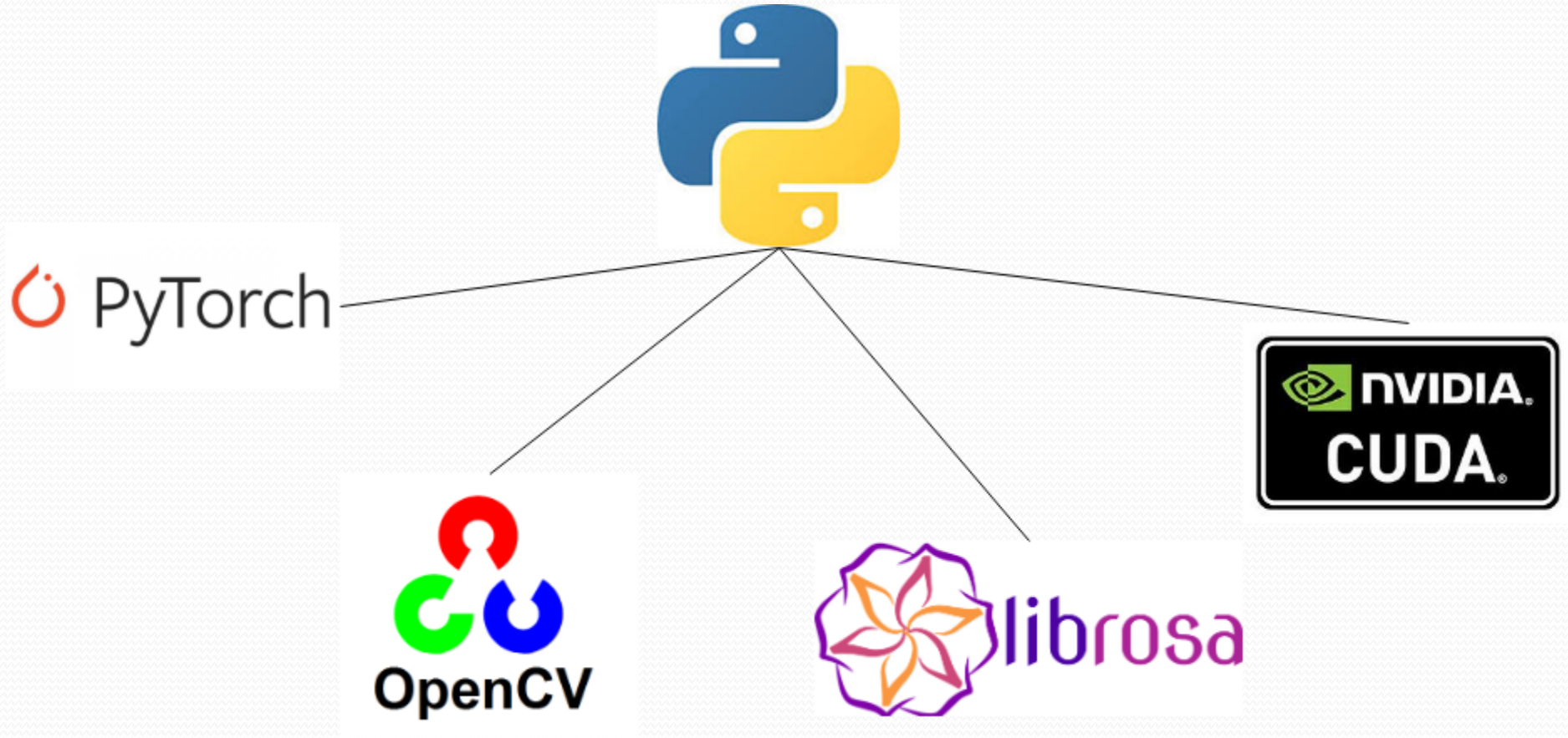
Reţea neurală tradiţională



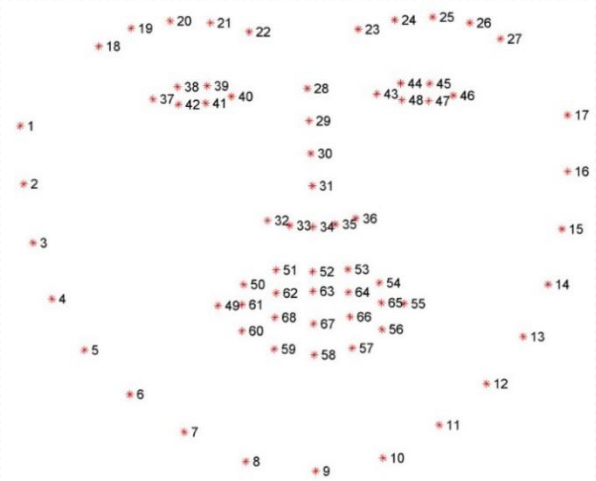
Reţea neurală recurentă



Tehnologii folosite

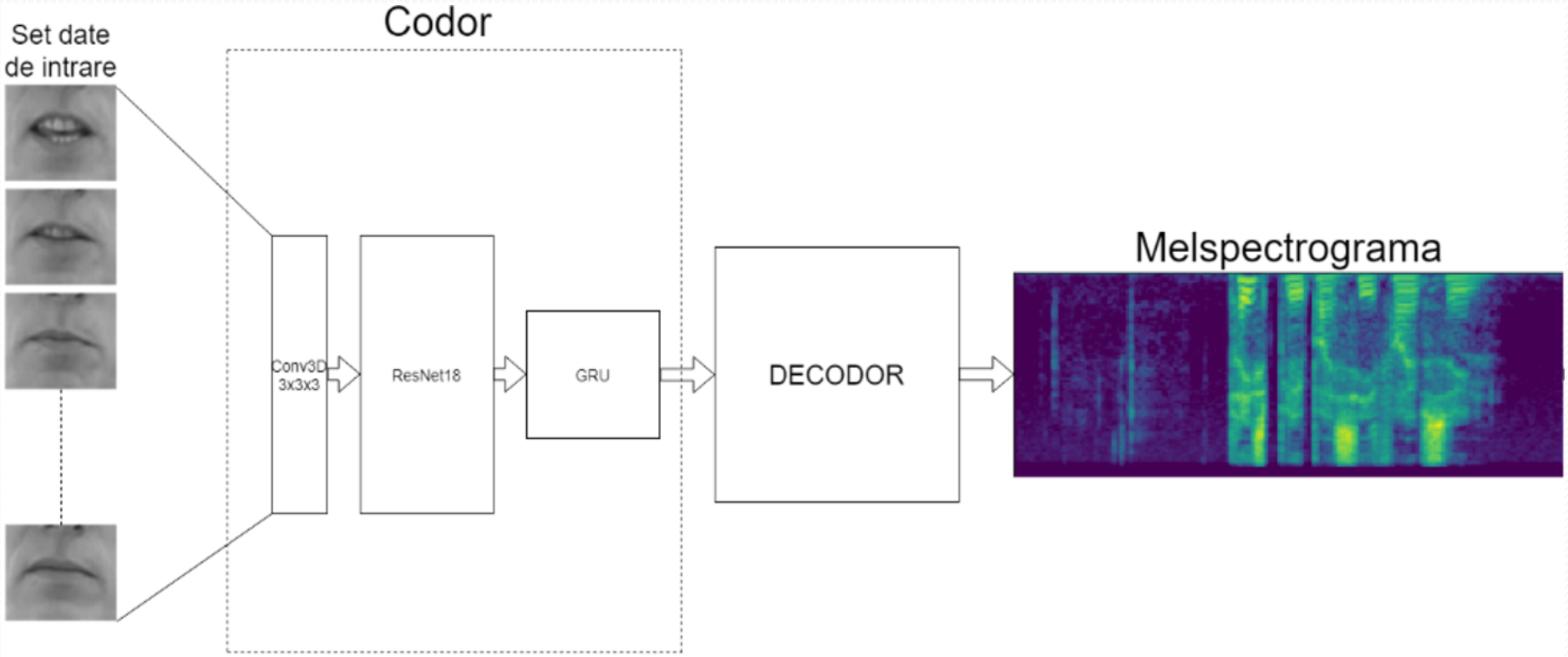


Prelucrarea imaginilor



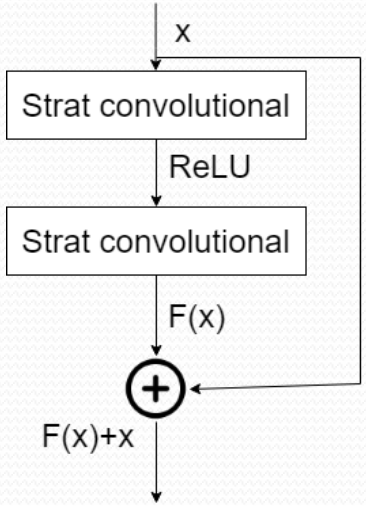
*model preantrenat de detectie feţe Sursa: [1]

Arhitectura folosită



ResNet18

Conexiune reziduală



Structura ResNet18

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

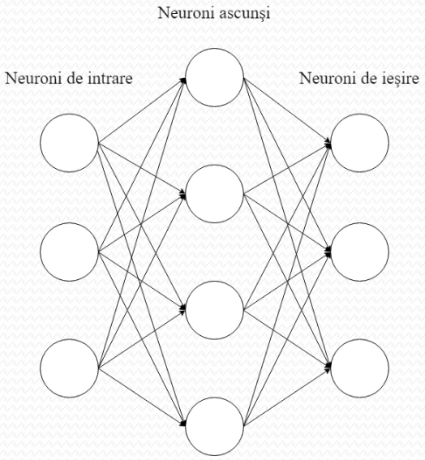
Implementare software

```
168         self.encoder = resnet18()
169         self.encoder.conv1 = nn.Conv2d(
170             K,
171             64,
172             kernel_size=(7, 7),
173             stride=(2, 2),
174             padding=(3, 3),
175             bias=False,
176         )
177
178         self.encoder = nn.Sequential(*list(self.encoder.children())[:-1])
```

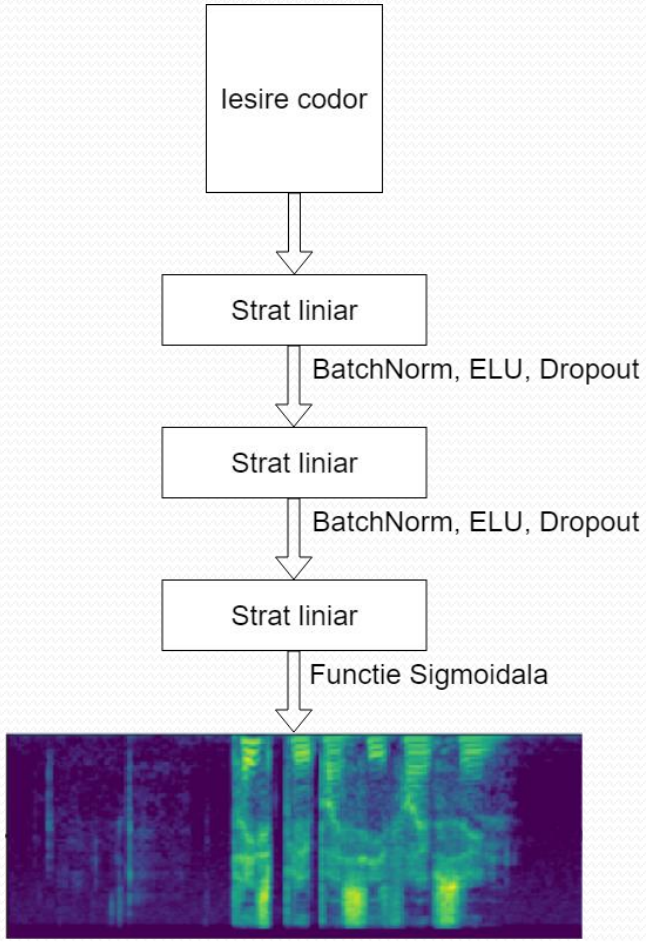
*Resnet18 Sursa:[2]

Decodor MLP

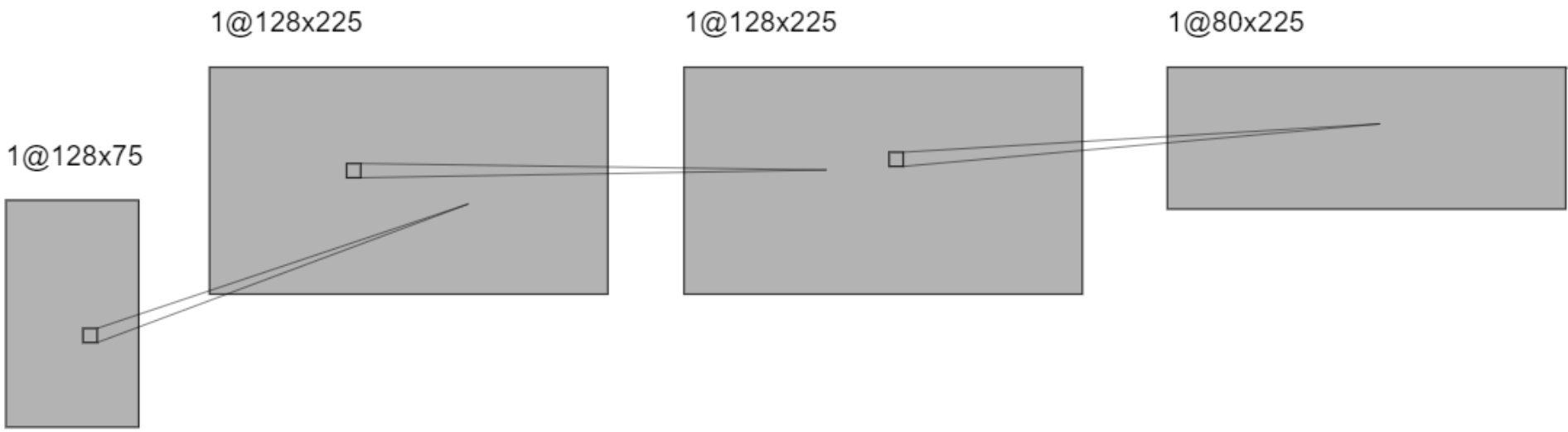
Strat liniar MLP



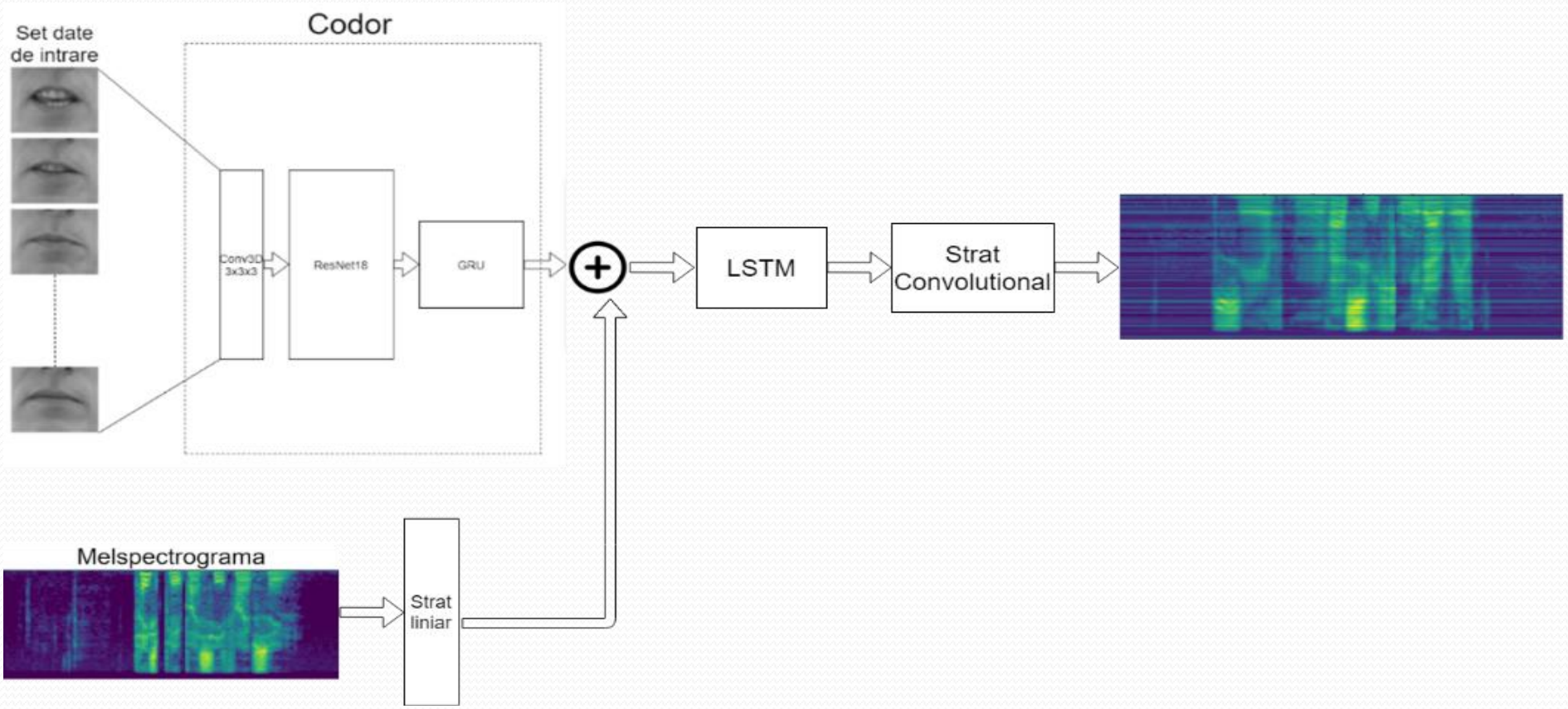
Structura decodorului



Decodor convoluţional



Decodor “Flowtron”



*Flowtron Sursa:[3]

Prelucrarea datelor pentru imbunătăţirea rezultatelor

- Padding
- Derivate de ordinul 1 si 2
- Normalizare

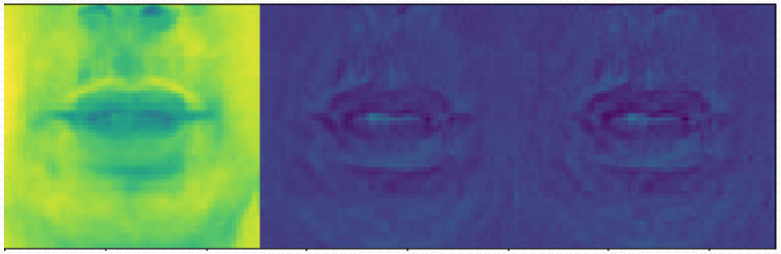
Padding

Eroarea medie patrată:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} (y - \hat{y}_i)^2$$

num. subjects	padding	first and second derivatives	normalization	decoder	sampling frequency	MSE ↓
1	10	yes	standardization	MLP	22 KHz	0.0058
1	25	yes	standardization	MLP	22 KHz	0.0059

Derivate de ordinul 1 si 2



num. subjects	padding	first and second derivatives	decoder	MSE ↓
1	10	no	MLP	0.00575
1	10	yes	MLP	0.00577
2	10	no	MLP	0.0056
2	10	yes	MLP	0.0053
5	10	no	MLP	0.0056
5	10	yes	MLP	0.0053

num. subjects	padding	first and second derivatives	decoder	MSE ↓
1	10	no	conv	0.0052
1	10	yes	conv	0.0052
2	10	no	conv	0.0053
2	10	yes	conv	0.0052

Normalizare

Formula folosită:
Rezultat = (intrare – medie) / abaterea medie standard

num. subjects	padding	first and second derivatives	normalization	decoder	sampling frequency	MSE ↓
2	10	yes	[0, 1]	MLP	22 KHz	0.0060
2	10	yes	standardization	MLP	22 KHz	0.0053

Implementare Software:

```
57      for data_, _ in train_loader:
58          batch_samples = data.size(0)
59          data = data.view(batch_samples, data.size(1), -1)
60          mean += data.mean(2).sum(0)
61          std += data.std(2).sum(0)
62          nb_samples += batch_samples
63
64      mean /= nb_samples
65      std /= nb_samples
```

Rezultate finale

num. subjects	padding	first and second derivatives	normalization	decoder	sampling frequency	MCD ↓
1	10	yes	standardization	MLP	22 KHz	183
1	10	yes	standardization	conv	19.3 KHz	169
1	10	yes	standardization	autoregressive	19.3 Khz	182

num. subjects	padding	first and second derivatives	normalization	decoder	sampling frequency	MCD
2	10	yes	standardization	conv	19.3 KHz	169
2	10	yes	standardization	autoregressive	19.3 Khz	191
2	10	yes	standardization	MLP	22 KHz	195

Mel Cepstral Distortion:

$$\frac{10\sqrt{2}}{\ln 10}\frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{i}\left(C_{ti}-\hat{C}_{ti}\right)^2}$$

Contribuţii personale

- Preprocesarea datelor
- Stabilirea arhitecturii generale
- Stabilirea si implementarea codorului si a decodoarelor
- Implementarea metodelor de procesare a datelor pentru imbunatăţirea rezultatelor
- Modificarea hiperparametrilor pentru potrivirea complexităţii arhitecturilor cu setul de date de intrare

Concluzii

- Scop inițial atins, reproducerea vocii este inteligibilă pentru:
  - Decodor convoluțional
  - Decodor autoregresiv
- Dezvoltari ulterioare
  - Integrarea posibilitații de a lucra in timp real
  - Identificarea emoțiilor si influențarea vocii pe baza lor

Demonstrarea funcţionalităţii

Audio original

Decodor MLP:

Decodor convolutional:

Decodor autoregresiv:

Cuvinte: “Set white at v1 again”

Bibliografie

- [1] J. Wang, K. Sun, T. Cheng, B. Jiang. Deep High-Resolution Representation Learning for Visual Recognition. 2020

- [2] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. Microsoft Research, 2015

- [3] R. Valle, K. Shih, R. Prenger, B. Catanzero. Flowtron: an autoregressive Flow-based Generative Network for Text-to-Speech Synthesis. 2020