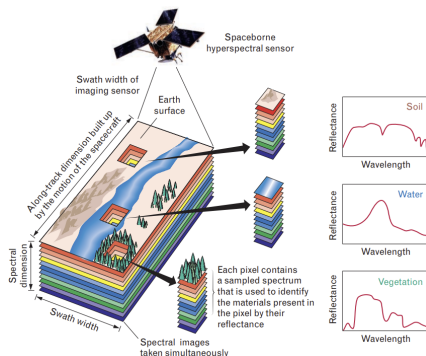


# Segmentarea semantică a imaginilor hiperspectrale

Bolohan Marian-Cristian  
333AB

# Imaginile hiperspectrale

**Imaginile hiperspectrale** sunt un tip special de imagini care captează informații detaliate despre **spectrul electromagnetic** al luminii reflectate sau emise de un obiect sau o scenă. Aceste imagini sunt colectate în mod special de sateliți și pot deveni una dintre cele mai importante unelte în *agricultură*, *predicția vremii* sau *prevenirea dezastrelor naturale*.



Din cauza calității proaste de vizualizare de către ochiul uman, aceste imagini trebuie **preprocesate**, **clasificate** și ulterior **segmentate** pentru a putea descoperi informații prețioase despre mediu.

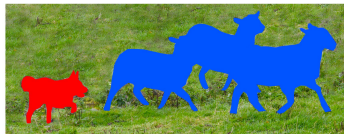
# Segmentarea imaginilor

**Segmentarea imaginilor** este o tehnică folosită în **computer vision** pentru a extrage „partiții” dintr-o imagine care aparțin unei anumite clase: *segmente de imagini*.

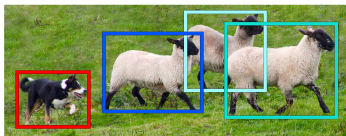
Termenul de segmentare de imagine este adesea confundat cu cel de *detecrie de obiecte* sau *clasificare de obiecte*, acestea fiind task-uri separate, care sunt tratate în mod diferit.



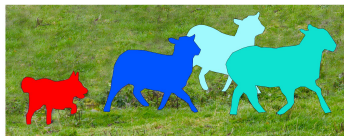
Image Recognition



Semantic Segmentation



Object Detection

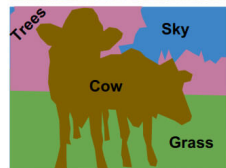
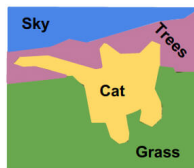


Instance Segmentation

# Segmentarea semantică

Astfel, prin **segmentare semantică** ne referim la o *clasificare pixel cu pixel*, fiecărui pixel din imagine îi este asociată o clasă.

Acest task este utilizat în foarte multe domenii precum: în *medicină*, la detecția tumorilor sau în *automotive*, la detecția indicatoarelor rutiere sau a obstacolelor din drum.



# Definirea seturilor de date

Pentru a testa performanța modelului realizat, am utilizat 3 seturi de date cu imagini hiperspectrale proveniți de la senzorul AVIRIS:

- **Salinas** este o vedere aeriană a unui câmp agricol din California
- **Indian Pines** este o panoramă a unui câmp agricol din Indiana
- **Pavia Center** este o imagine achiziționată de senzorul ROSIS a unui oraș din nordul Italiei



Figura: Salinas

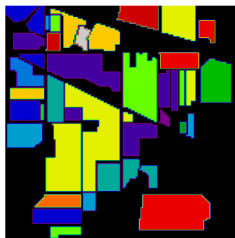


Figura: Indian Pines

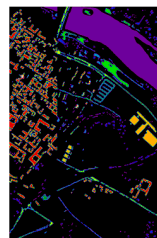


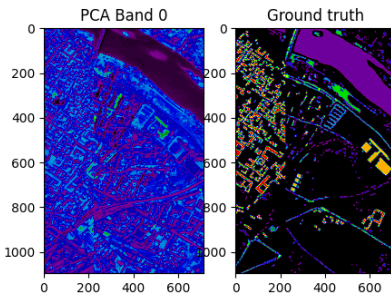
Figura: Pavia

# Reducere dimensională: PCA

Dintre toate metodele de reducere dimensională, pentru imaginile hiper-spectrale cea mai eficientă s-a dovedit a fi **PCA (Principal Component Analysis)**, prin timpul de execuție mic, dar și puterea mare de reconstrucție folosind puține componente principale.

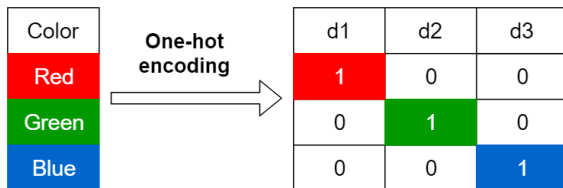
Pentru seturile de date folosite, prin alegerea  $r = 10$  componente principale am recuperat peste 95% din toată informația:

Setul de date	Informația recuperată
Salinas	99.96%
Indian Pines	96.91%
Paiva Center	99.85%



# Preprocesări ulterioare și one-hot encoding

Pentru a putea folosi un singur model pentru toate seturile de date, indiferent de dimensiunile imaginilor, redimensionăm imaginile la 144x144 pixeli. Pentru a putea clasifica și ulterior segmenta imaginile, creăm o matrice de etichete pentru fiecare pixel din imagine, folosind **One-Hot Encoding**. Acesta crează câte o imagine pentru fiecare clasă, unde valoarea 1 a unui pixel reprezintă apartenența sa la acea clasă, iar valoarea 0 reprezintă apartenența la o clasă diferită de acesta.



# Preprocesări ulterioare și one-hot encoding

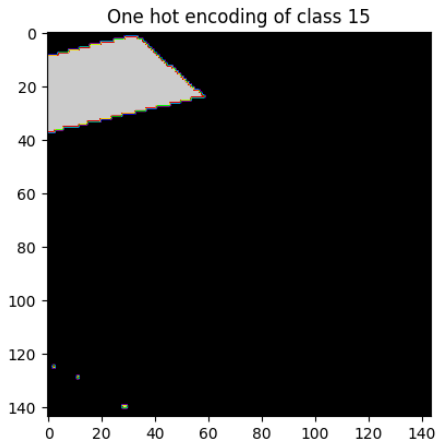
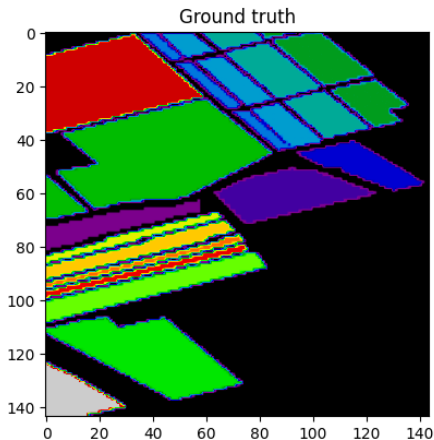
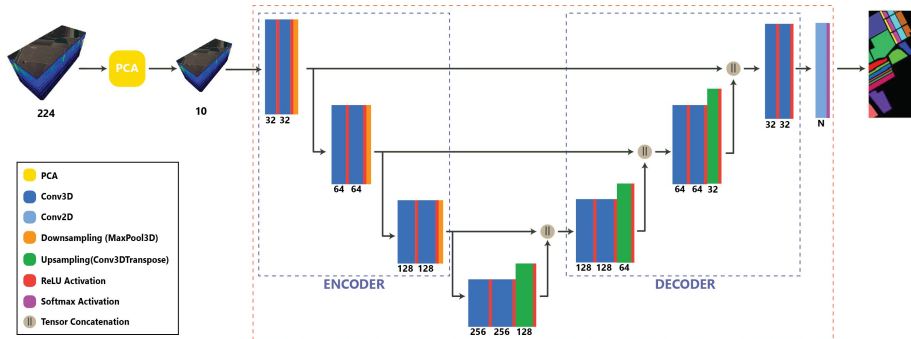


Figura: One-hot encoding pentru setul de date *Salinas*



# Definirea arhitecturii modelului

Pentru a segmenta eficient imaginile hiperspectrale am folosit un model de **rețele neuronale convoluționale (CNN)**. Rețeaua folosită are la bază arhitectura **U-NET**, o rețea de tip *encoder-decoder*, care nu are straturi complet conectate, făcând-o deci, un model pur convoluționar.



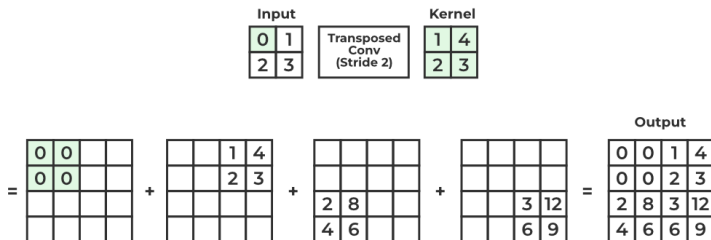
Acestă rețea este alcătuită din:

- **Rețeaua Encoder:** Are rolul de a extrage *feature-uri* și de a învăța diverse *pattern-uri* ale datelor. Este alcătuită din 3 blocuri care conțin câte 2 straturi convoluționale **Conv3D**, activate folosind funcția **ReLU**. Aceste straturi sunt folosite pentru a învăța nu doar informații din diversele benzi spectrale, ci și anumite relații dintre acestea. În final, urmează un strat **MaxPool3D** pentru a reduce dimensiunile imaginilor la jumătate, reducând astfel numărul parametrilor de antrenat.
- **Rețeaua Decoder:** Acesta primește reprezentarea abstractă de la Encoder și generează masca de segmentare. Acesta are structură simetrică cu Encoder-ul, fiind formată din aceleași straturi convoluționale, însă nu mai face *downsampling*, folosind MaxPool, ci face *upsampling*, folosind **convoluția transpusă (Conv3DTranspose)**.

# Convoluția Transpusă

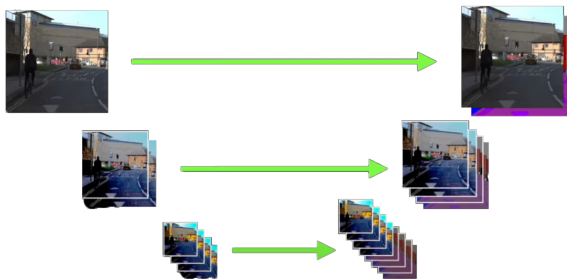
**Convoluția transpusă** este o operație de *upsampling*, asemănătoare convoluției prin care, folosind kernel-uri adecvate, rezultă matrici de dimensiuni mai mari decât cea inițială, invers operației clasice.

Pentru a reconstrui masca de segmentare, se folosește *stride=2* pentru a dubla dimensiunea imaginii inițiale, care a fost înjumătățită folosind Max Pool în Encoder.



# Bridge și Skip connections

- Zona **Bridge** face legătura între Encoder și Decoder, completând transferul în formă de **U** al informației. Acesta este alcătuit din 2 starturi convoluționale, activate folosind ReLU.
- Conexiunile **Skip** reprezintă motivul pentru care aceste rețele sunt atât de eficiente. Acestea transmit direct informația de la Encoder la Decoder, în mod simetric. De asemenea, sunt folosite pentru pentru a perfoma *backpropagation* mai eficient, ajutând rețeaua să învețe reprezentări mai bune.



# Antrenarea modelului și definirea hiperparametrilor

Implementarea rețelei convoluționale a fost realizată folosind Tensorflow. Pentru antrenarea rețelei au fost folosiți optimizatorul *ADAM*, iar ca hiperparametrii: *learning\_rate* = 0.001, *batch\_size* = 1, și numărul de epoci variabil, pentru a evalua rețeaua.

Ca și funcție de loss, acesta fiind la bază o problemă de clasificare, am folosit *cross entropy loss*:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(\hat{y}_{ij})$$

unde:  $N$  reprezintă numărul de exemple,  $C$  este numărul de clase,  $y_{ij}$  este eticheta pentru clasa  $j$  a exemplului  $i$ , iar  $\hat{y}_{ij}$  este probabilitatea prezisă pentru clasa  $j$  a exemplului  $i$ .

# Evaluarea modelului și metricile utilizate

Pentru evaluarea modelului de segmentare, am folosit câteva dintre metricile uzuale, precum *acuratețe* și *precizie*, dar și *indicele Jaccard*, numit și **IoU (intersection over union)**.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$precision = \frac{TP}{TP + FP}$$

$$accuracy = \frac{TP}{TP + FN}$$

$$F1 = \frac{2TP}{2TP + FP + FN}$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

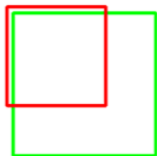
Figura: Matricea de confuzie

# IoU (Intersection over Union)

**IoU** este o metrică foarte importantă în evaluarea modelelor de segmentare, deoarece cuantifică cât de bine face diferența între obiecte și fundal. După cum îi sugerează numele, acesta se calculează prin raportul intersecției dintre masca de segmentare și masca obținută și reuniunea acestora:

$$IoU = \frac{A \cap B}{A \cup B} = \frac{TP}{TP + FP + FN}$$

IoU: 0.4034



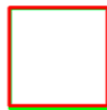
Poor

IoU: 0.7330



Good

IoU: 0.9264



Excellent

# Rezultate: *Salinas*

Nr. epoci	IoU	Accuracy	Pecision	Recall	F1	Timp de antrenare
200	77.35%	92.64%	87.85%	86.44%	86.94%	7min
250	88.32%	97.70%	95.38%	94.87%	95.07%	9min
300	100%	100%	100%	100%	100%	11min

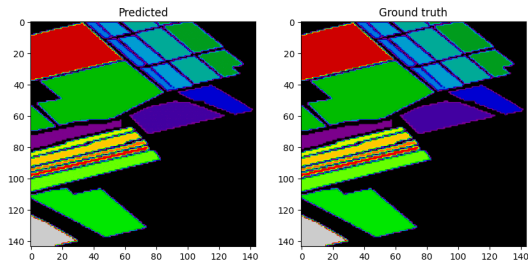


Figura: Masca de segmentare obținută

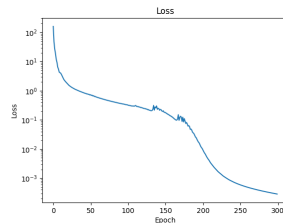


Figura: Loss pentru 300 epoci



# Rezultate: *Indian Pines*

Nr. epoci	IoU	Accuracy	Pecision	Recall	F1	Timp de antrenare
200	62.80%	89.16%	81.49%	78.79%	79.63%	7min
250	82.27%	96.50%	94.60%	92.75%	93.35%	9min
300	98.35%	99.97%	99.91%	99.93%	99.92%	11min

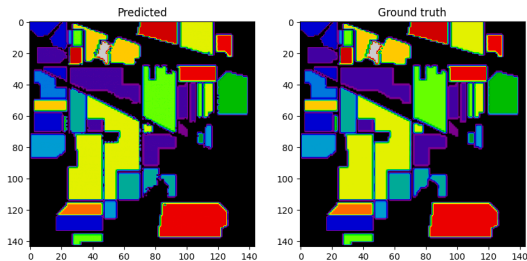


Figura: Masca de segmentare obținută

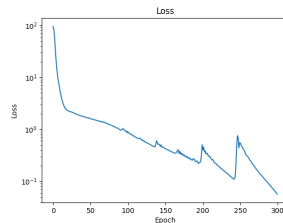


Figura: Loss pentru 300 epoci

# Rezultate: *Pavia Center*

Nr. epoci	IoU	Accuracy	Pecision	Recall	F1	Timp de antrenare
250	38.72%	86.01%	55.26%	34.90%	42.78%	9min
300	46.77%	92.42%	86.70%	52.00%	61.99%	11min
350	98.32%	99.99%	99.99%	99.99%	99.99%	13min

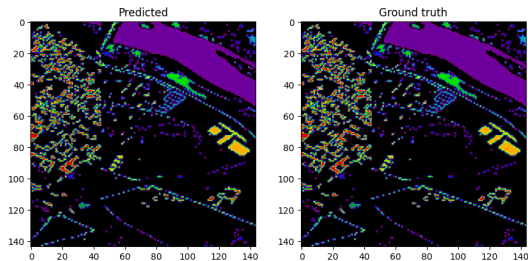


Figura: Masca de segmentare obținută

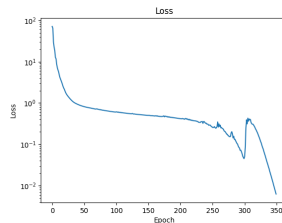


Figura: Loss pentru 350 epoci

- 1 Dataset-uri: [www.ehu.eus](http://www.ehu.eus)
- 2 Jiaju Li; Hefeng Wang; Anbing Zhang Yuliang Liu *Semantic Segmentation of Hyperspectral Remote Sensing Images Based on PSE-UNet Model*
- 3 J Nishchal; Sanjana Reddy; N Navya Priya; Varsha R Jenni; R Hebbar; B. Sathish Babu *Pansharpening and Semantic Segmentation of Satellite Imagery*
- 4 <https://www.superannotate.com/blog/guide-to-semantic-segmentation>
- 5 <https://medium.com/analytics-vidhya/what-is-unet-157314c87634>
- 6 [https://d2l.ai/chapter\\_computer-vision/transposed-conv.html](https://d2l.ai/chapter_computer-vision/transposed-conv.html)
- 7 Tensorflow documentation:  
[https://www.tensorflow.org/api\\_docs/python/tf/keras/Model](https://www.tensorflow.org/api_docs/python/tf/keras/Model)