

# Ionospheric Echo Detection Using Convolutional Neural Networks

César De La Jara

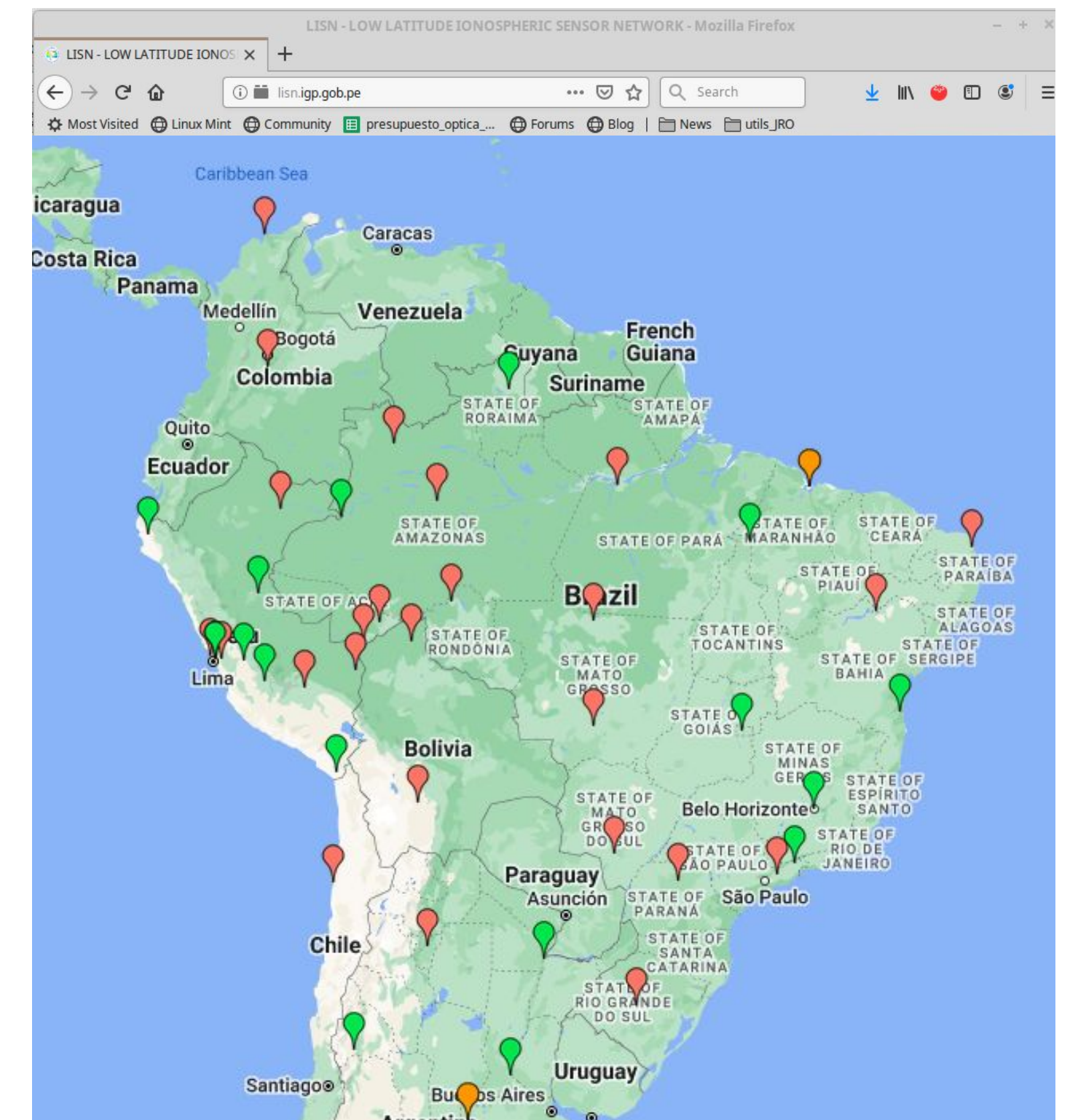
Radio Observatorio de Jicamarca – Instituto Geofísico del Perú

## LISN project

LISN is a multi instrument, multi institution, multi project where geophysical observation instruments have been deployed in several South American countries.

Study the electrodynamics of the ionosphere.

Data freely available at [www.lisn.igp.gob.pe](http://www.lisn.igp.gob.pe)



**LISN instruments**

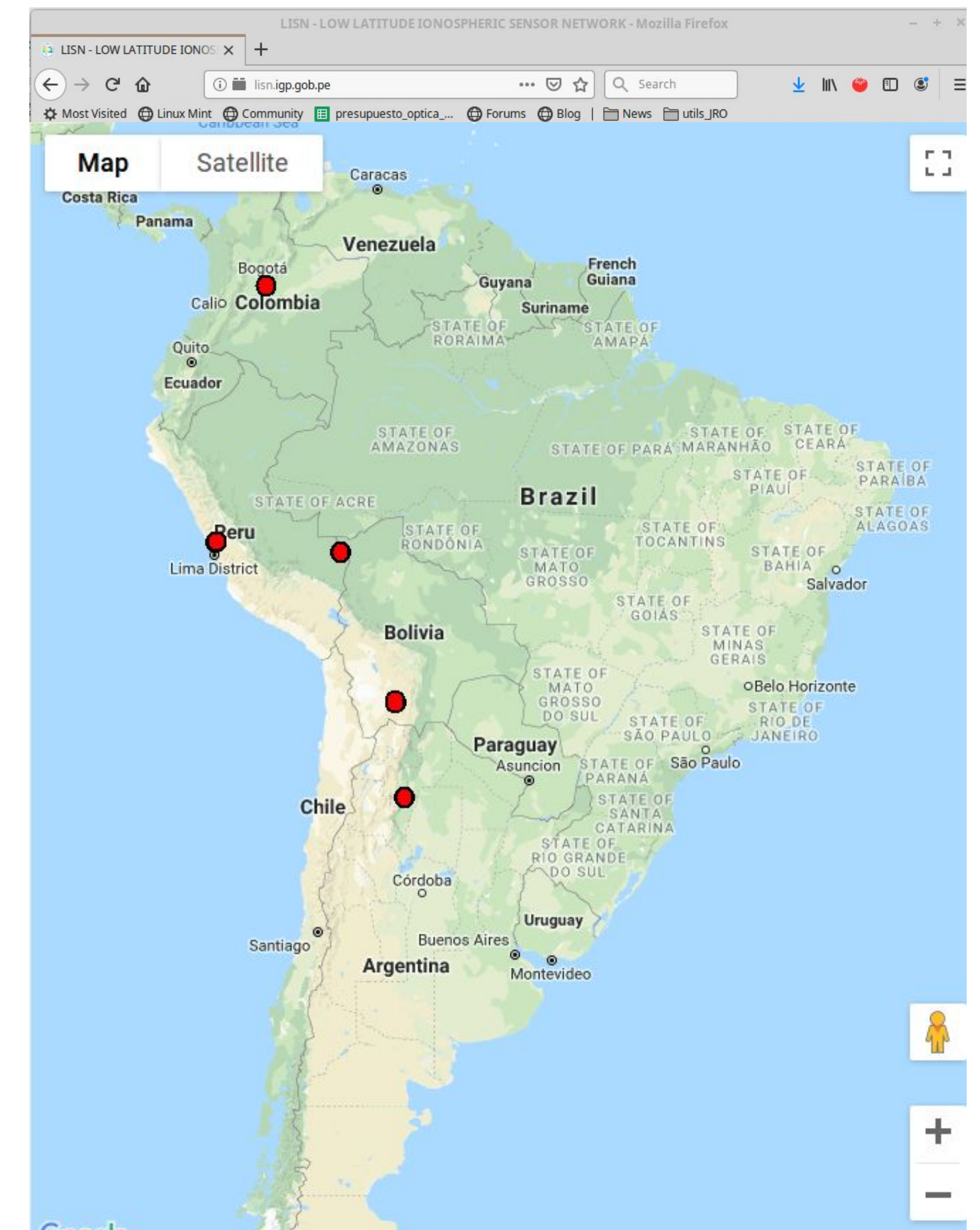


## Dataset

LISN has 4 ionosondes in Perú, Argentina and Bolivia. One will be installed in Colombia.

These ionosondes generate a big amount of data -> CNN to detect echoes

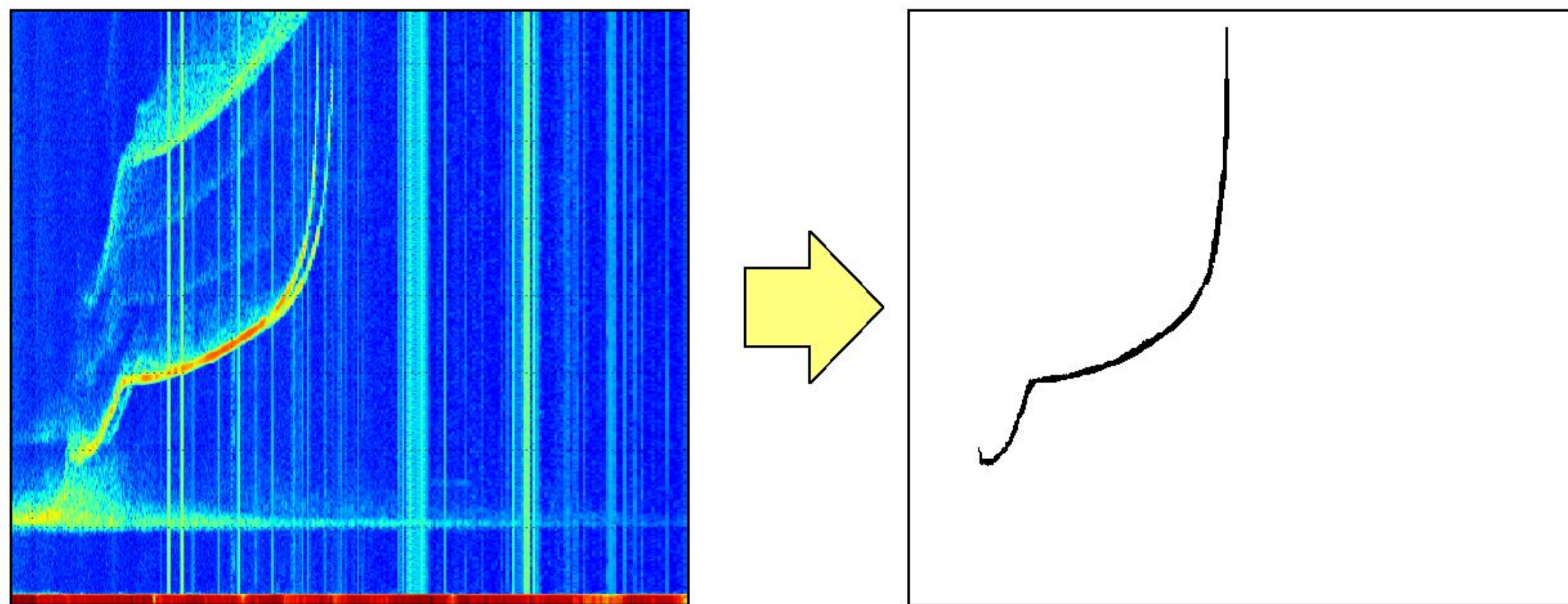
- 1,000,000 ionograms in the database.
- 50,780 ionograms used:
- 512 ranges x 408 frequencies
- 817 were selected for labeled data (manual echo extraction)
- <https://www.kaggle.com/cdelajara/ionograms>



LISN ionosondes

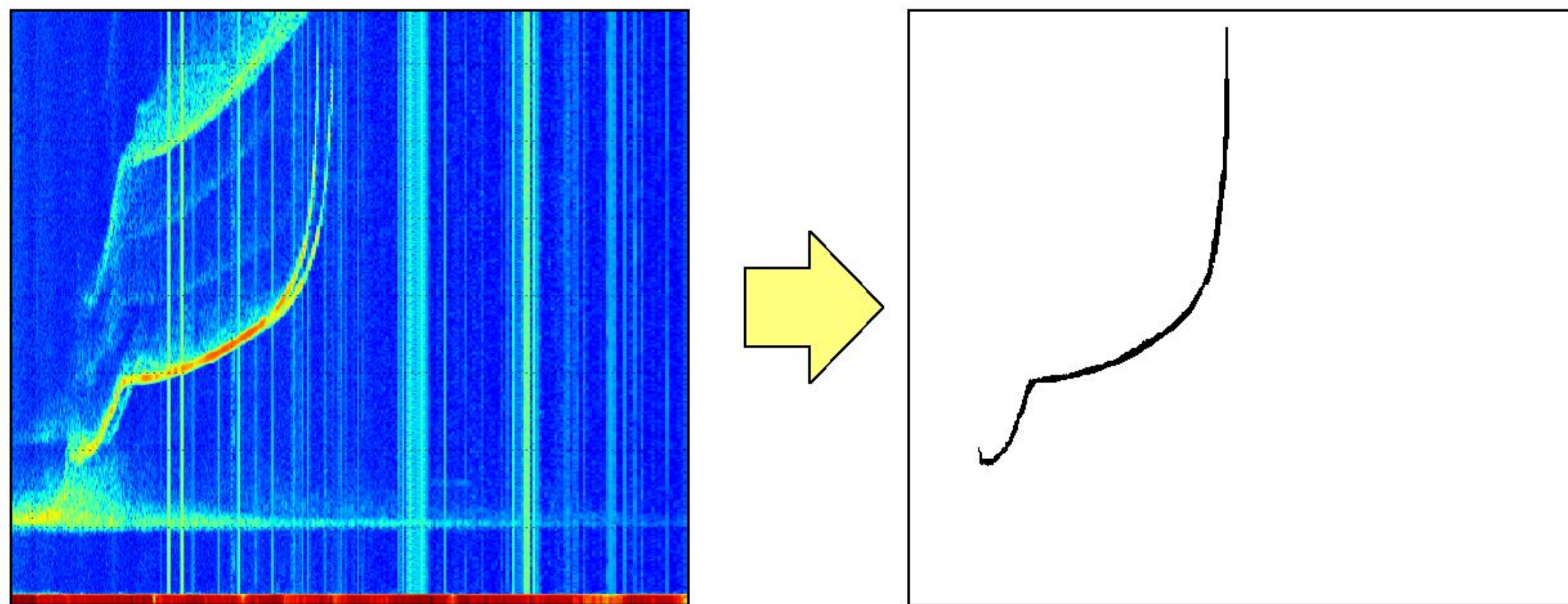


It is not possible to use manual scaling regularly → Deep learning model.  
This work is an attempt to develop a system to scale LISN ionograms.





We propose a method to extract ionospheric echoes from digital ionograms using neural networks implementing semantic segmentation, where the goal is to label each pixel as echo or background.

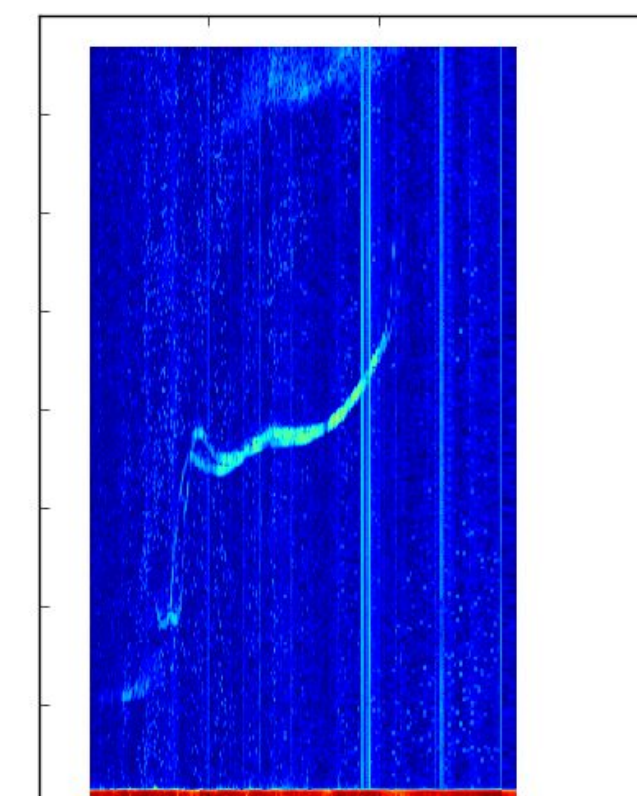


## Performance metrics:

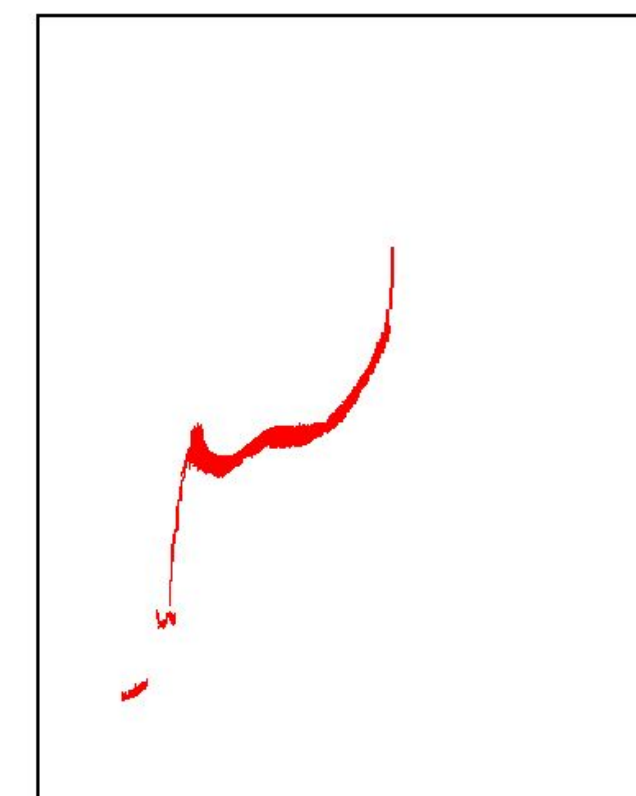
Extract ionospheric layers

Pixel: ionospheric trace or background

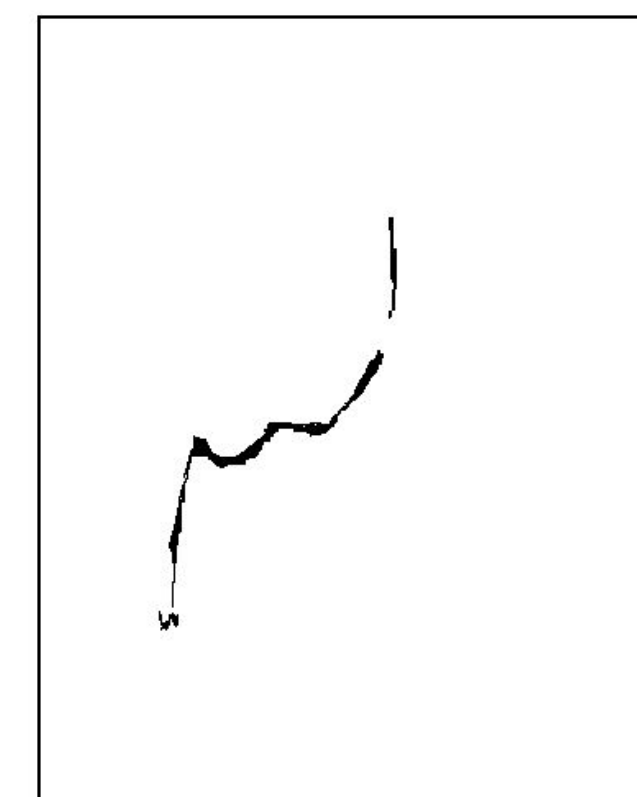
$$\text{IoU} = \frac{\text{common area}}{\text{total area}}$$



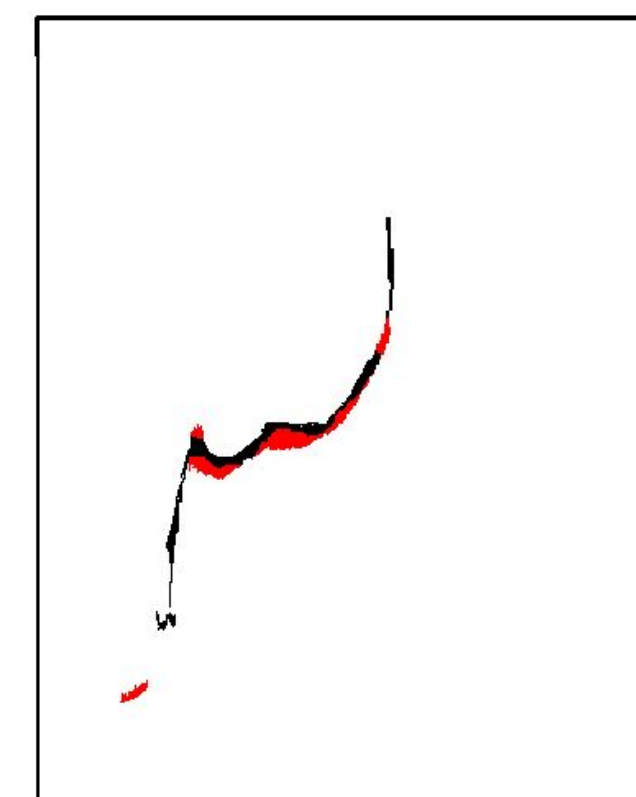
Ionograma Original



Segmentación manual



Segmentación automática



Determinación de la calidad  
con IoU



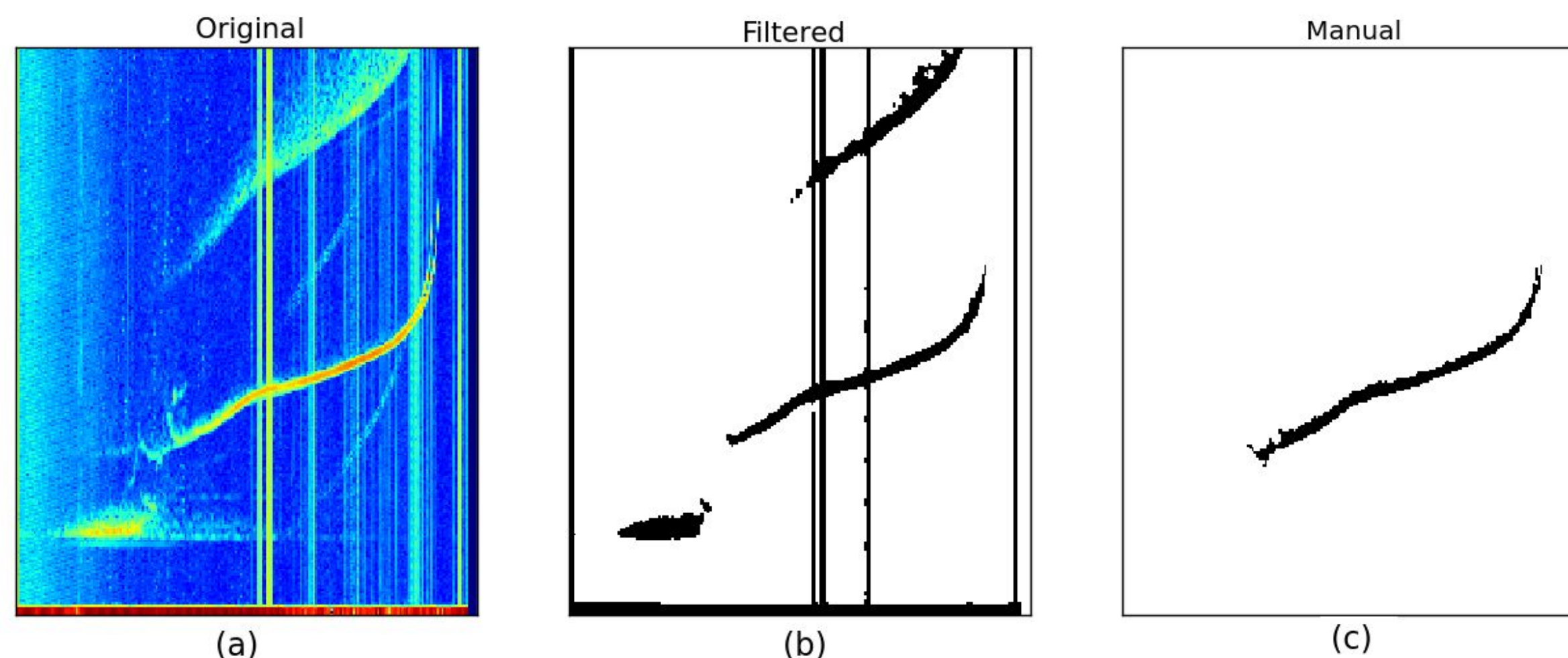
**1st stage:** Baseline models:

- Filtering and Thresholding
- Unsupervised machine learning models

**2nd stage:** Neural network models

**Filtering and thresholding:** Ionograms are considered as images in which the noise should be filtered out to isolate ionospheric echoes:

Median filter + Thresholding



Average IoU: 0.163

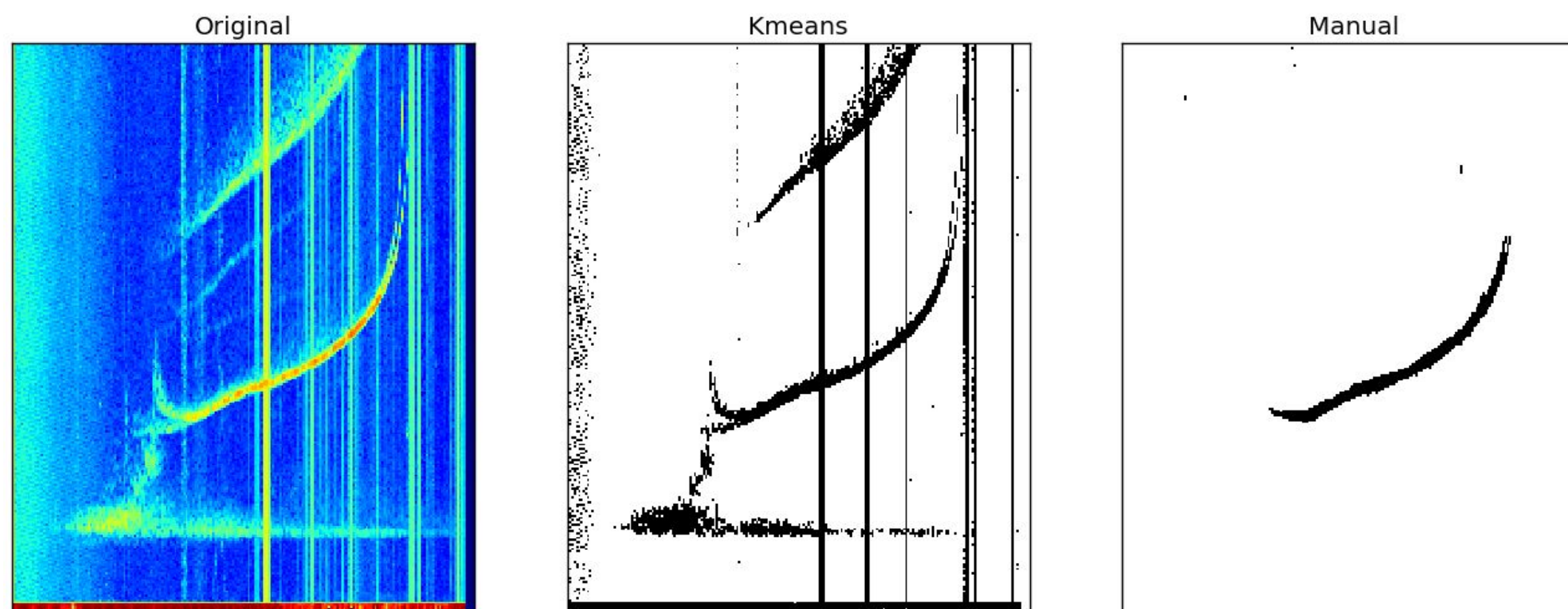


## Unsupervised machine learning models:

Ionograms are considered three dimensional arrays ( $x$ ,  $y$ ,  $V$ ).

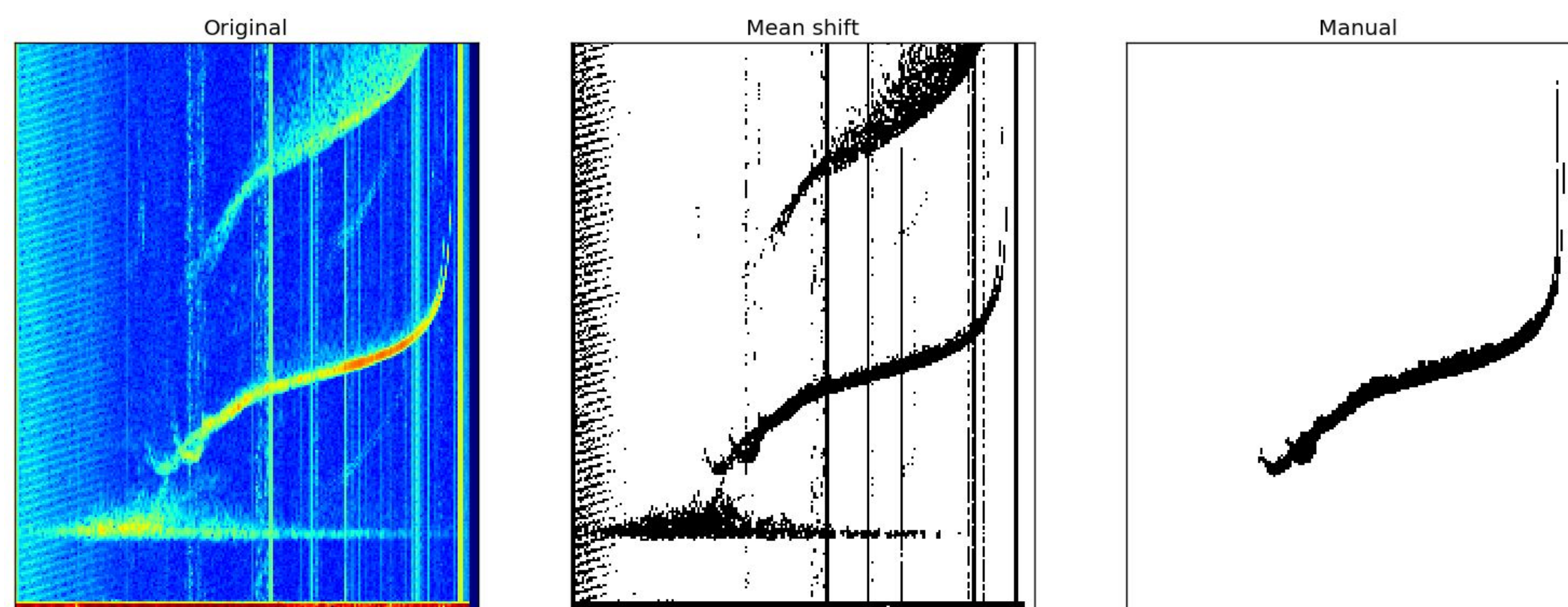
**K-Means:**  $K=2$ . Background and echo

Average IoU: 0.157



## Unsupervised machine learning models:

**Mean Shift:** It doesn't require a previous knowledge of the numbers of clusters



Average IoU: 0.105



## Profile detection using CNNs:

LISN database: big amount of unlabeled data

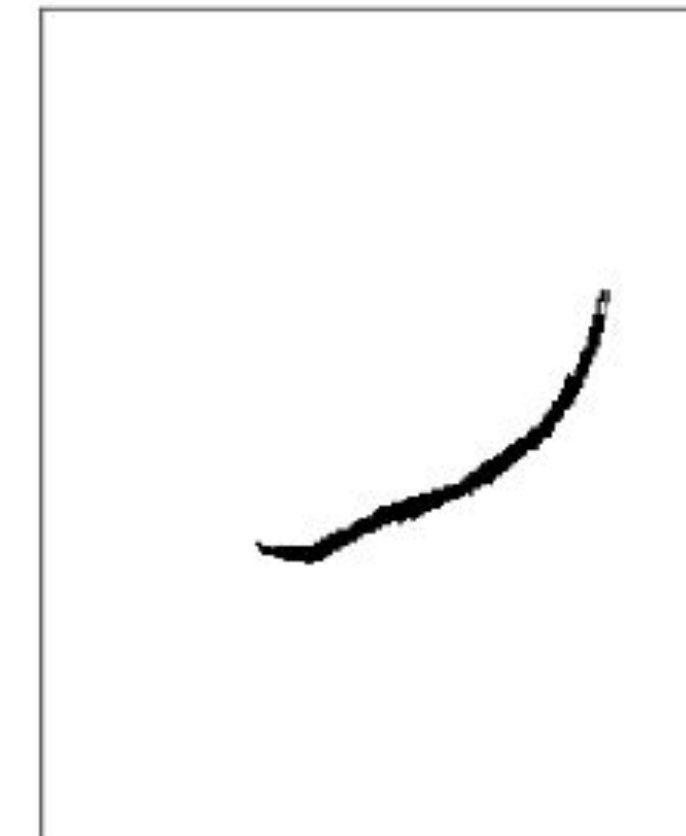
Results

«« « 1 2 3 4 5 » »»

Files: Oct 07, 2015

JM91J_2015280143804.ngi	JM91J_2015280144304.ngi	JM91J_2015280144803.ngi	JM91J_2015280145303.ngi
JM91J_2015280145803.ngi	JM91J_2015280150303.ngi	JM91J_2015280150803.ngi	JM91J_2015280151303.ngi
JM91J_2015280151803.ngi	JM91J_2015280152303.ngi	JM91J_2015280152803.ngi	JM91J_2015280153303.ngi
JM91J_2015280153803.ngi	JM91J_2015280154303.ngi	JM91J_2015280154803.ngi	JM91J_2015280155303.ngi
JM91J_2015280155803.ngi	JM91J_2015280160303.ngi	JM91J_2015280160803.ngi	JM91J_2015280161303.ngi
JM91J_2015280161803.ngi	JM91J_2015280162303.ngi	JM91J_2015280162803.ngi	JM91J_2015280163303.ngi
JM91J_2015280163803.ngi	JM91J_2015280164303.ngi	JM91J_2015280164803.ngi	JM91J_2015280165303.ngi
JM91J_2015280165803.ngi	JM91J_2015280170303.ngi	JM91J_2015280170803.ngi	JM91J_2015280171303.ngi
JM91J_2015280171803.ngi	JM91J_2015280172303.ngi	JM91J_2015280172803.ngi	JM91J_2015280173303.ngi
JM91J_2015280173803.ngi	JM91J_2015280174303.ngi	JM91J_2015280174803.ngi	JM91J_2015280175303.ngi
JM91J_2015280175803.ngi	JM91J_2015280180303.ngi	JM91J_2015280180803.ngi	JM91J_2015280181303.ngi
JM91J_2015280181803.ngi	JM91J_2015280182303.ngi	JM91J_2015280182803.ngi	JM91J_2015280183303.ngi
JM91J_2015280183803.ngi	JM91J_2015280184303.ngi	JM91J_2015280184803.ngi	JM91J_2015280185303.ngi
JM91J_2015280185803.ngi	JM91J_2015280190303.ngi	JM91J_2015280190804.ngi	JM91J_2015280191303.ngi
JM91J_2015280191803.ngi	JM91J_2015280192304.ngi	JM91J_2015280192803.ngi	JM91J_2015280193303.ngi

Labeled data: manual segmentation



## Semi supervised model:

- Large number of unlabeled data
- Reduced number of labeled data

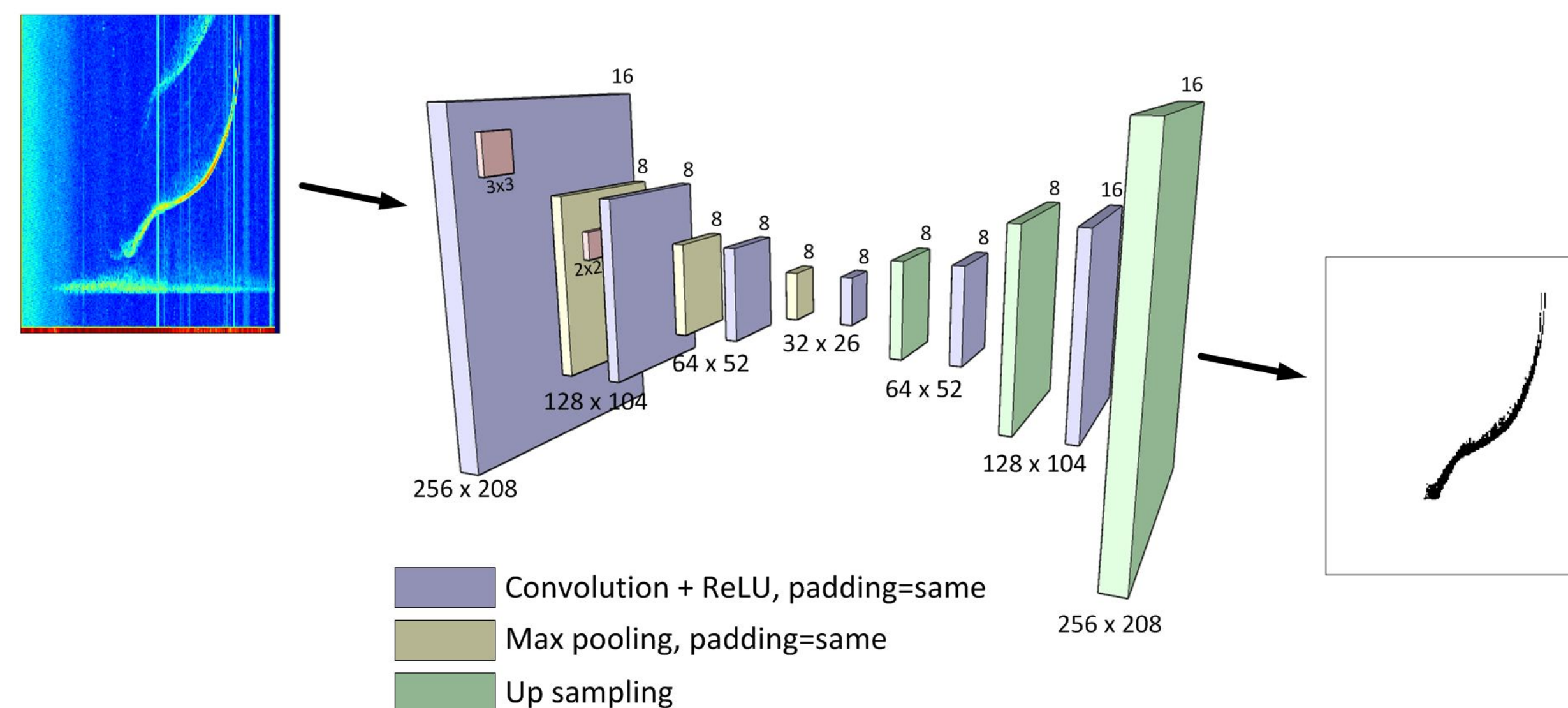
## Profile detection using CNNs:

Multilayer convolutional encoder decoder neural network, with multiple layers of convolutions.

The encoder maps the raw inputs to a representations of feature vectors

The decoder processes this features representation to produce an output in the same format than the raw input.

Efficient representation of the data.



- 6 convolution layers + ReLU
- 3 max pooling, zero padding
- 3 upsampling layers
- Binary cross entropy
- Ada delta



**Pre training:** Building a neural network model from baseline models

Input: original ionograms.

Output: echoes extracted using baseline models.

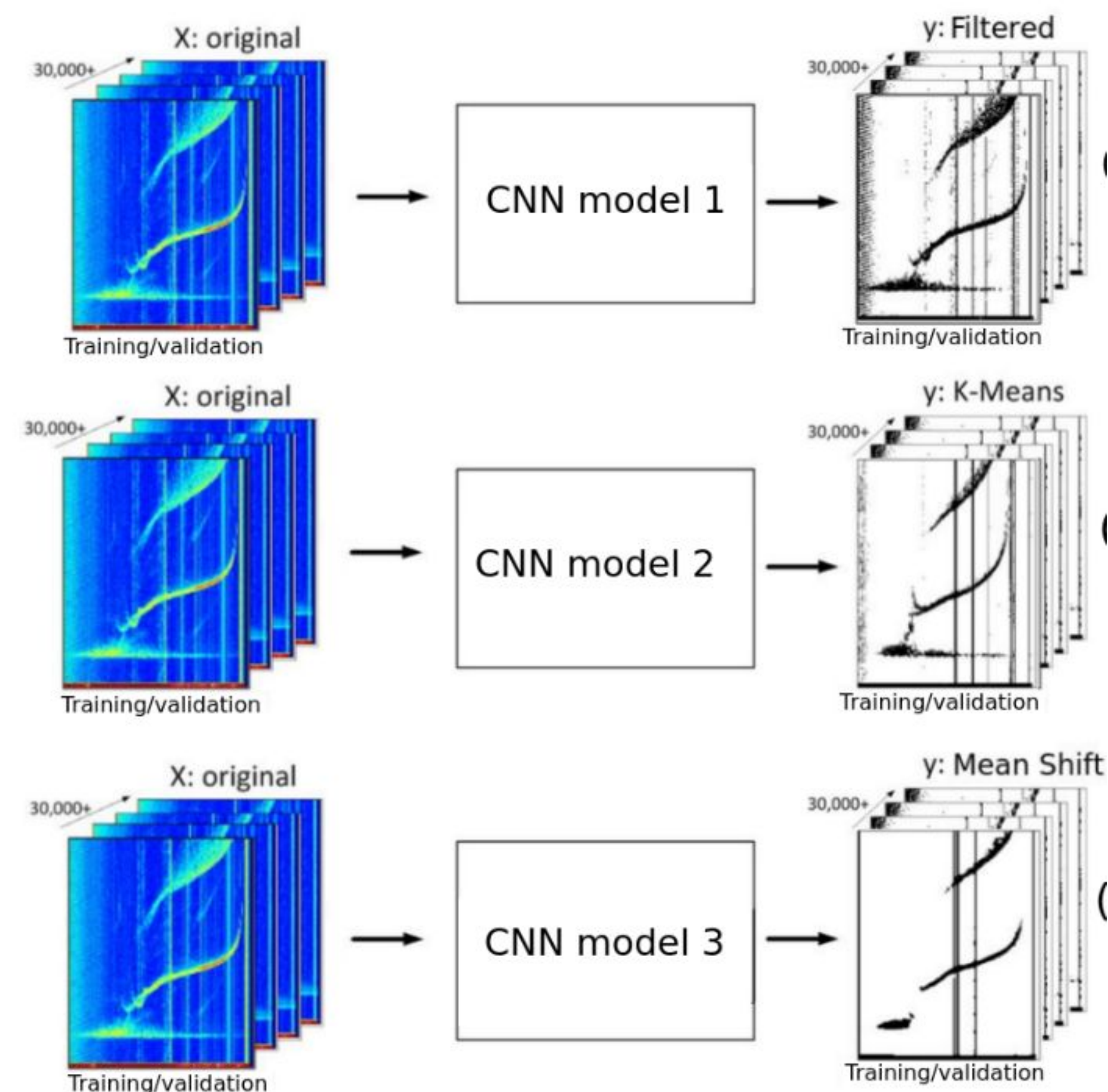
3 CNN models are created.

Average IoU:

Filtered: 0.174

K-Means: 0.173

Mean shift: 0.077



**Pre training:** Building a neural network model from baseline models

Input: original ionograms.

Output: echoes extracted using baseline models.

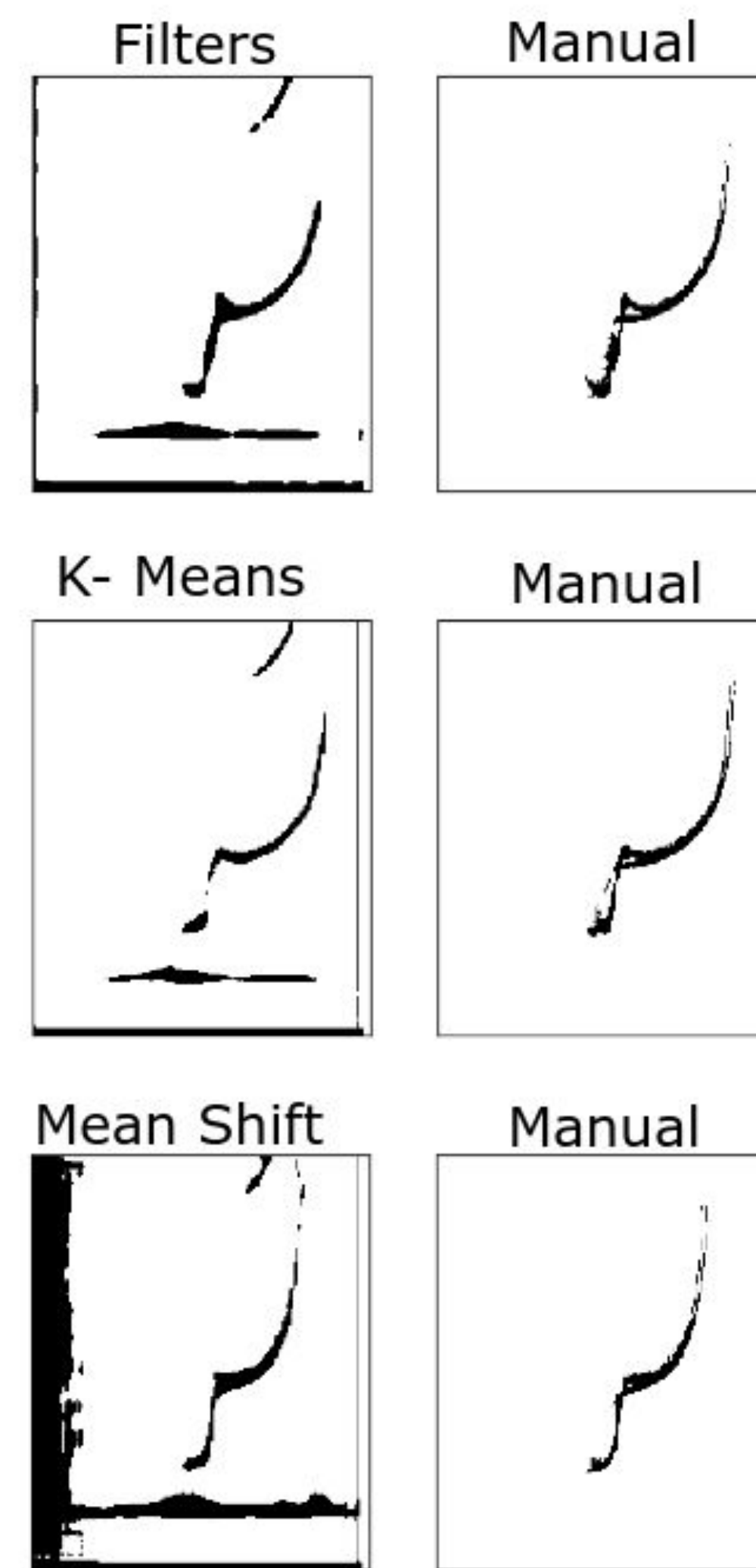
3 CNN models are created.

Average IoU:

Filtered: 0.174

K-Means: 0.173

Mean shift: 0.077





## Fine tuning

Input: original ionograms.

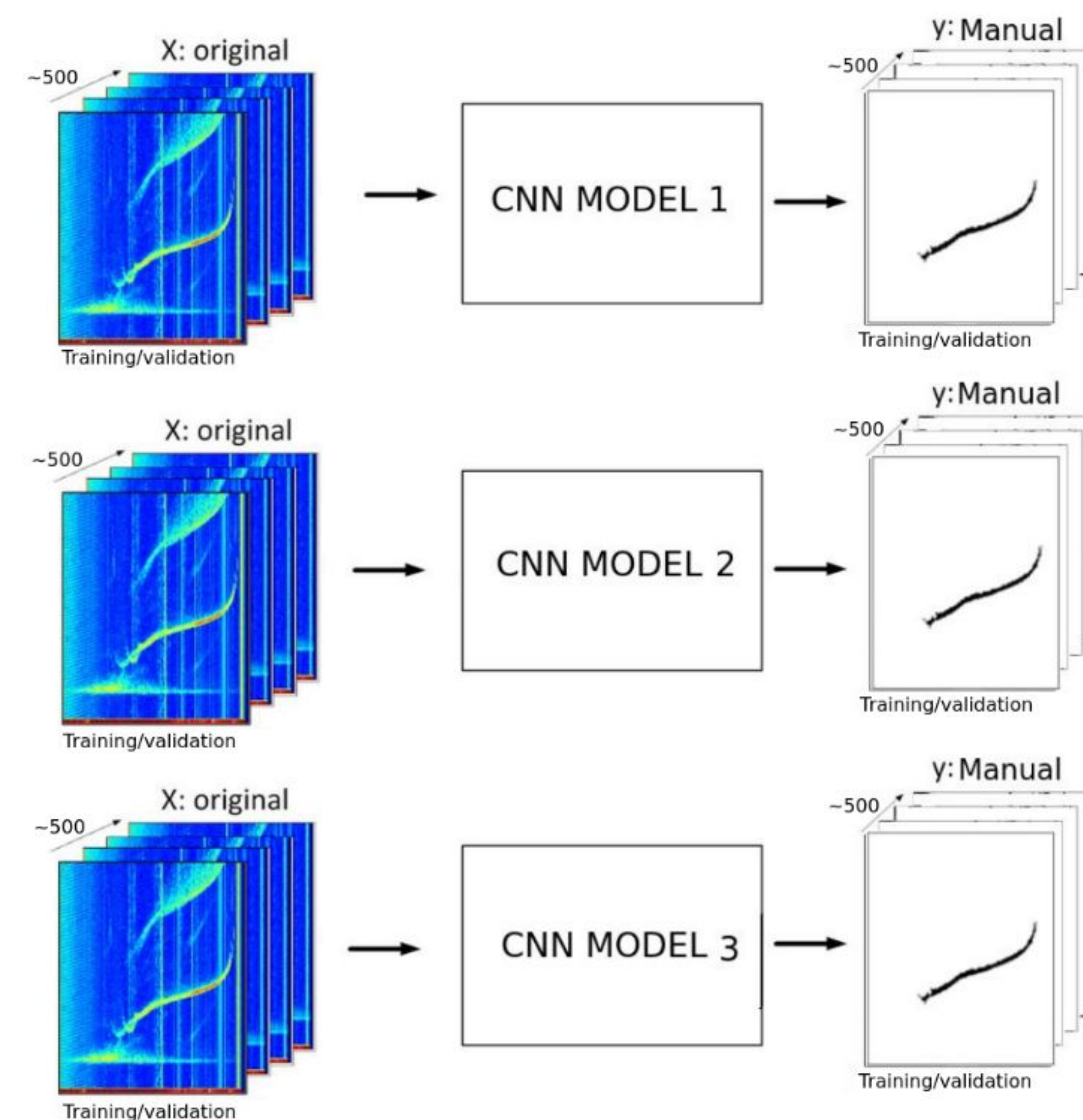
Output: echoes manually extracted.

Average IoU:

Filtered: 0.589

K-Means: 0.602

Mean shift: 0.593



## Fine tuning and final prediction:

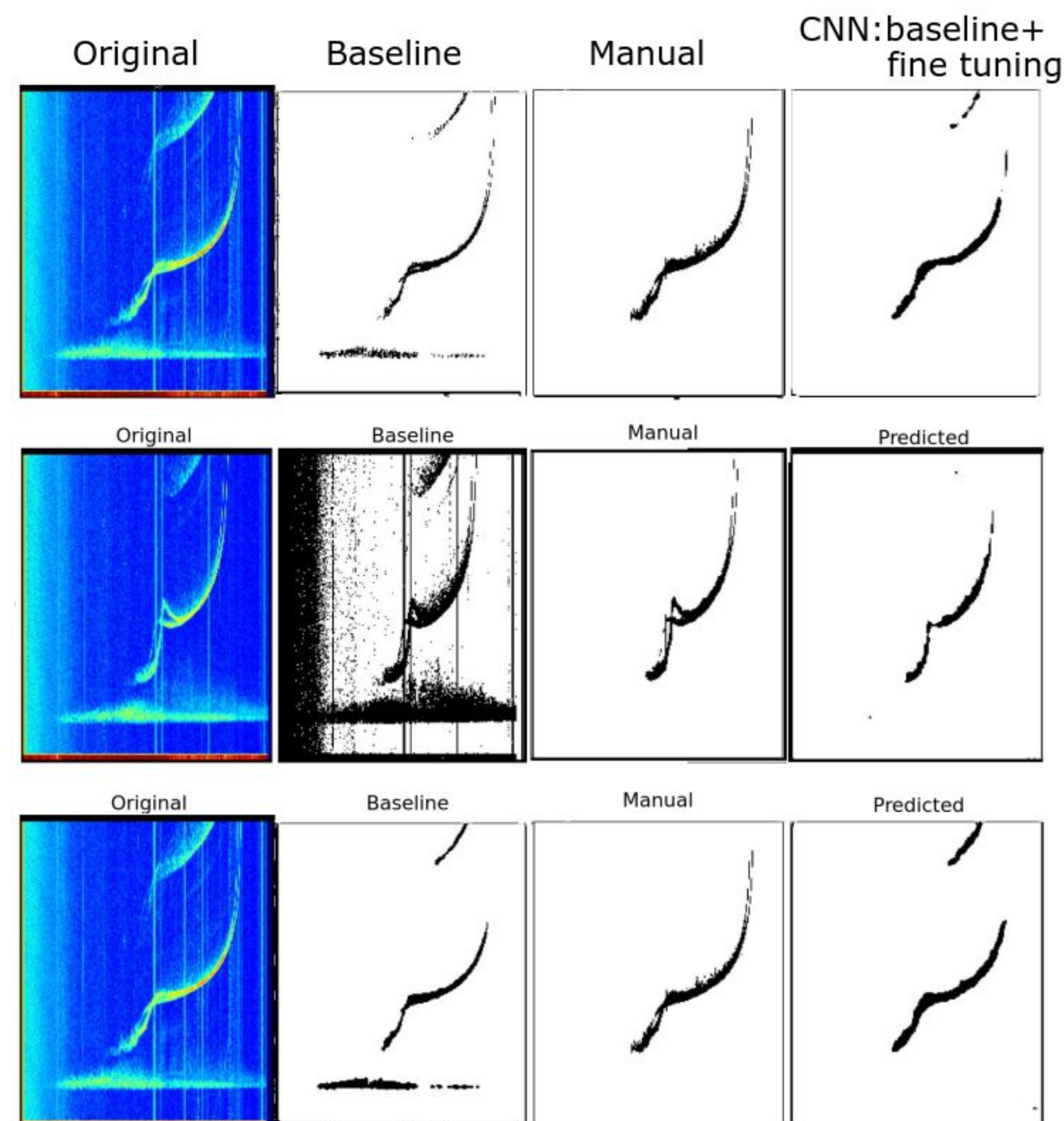
Results of the different stages of the training process

K-means

Mean shift

Filters

A final model was created using only manually extracted data.  
Average IoU of 0.569.





## Fine tuning and final prediction:

Results of the different stages of the training process

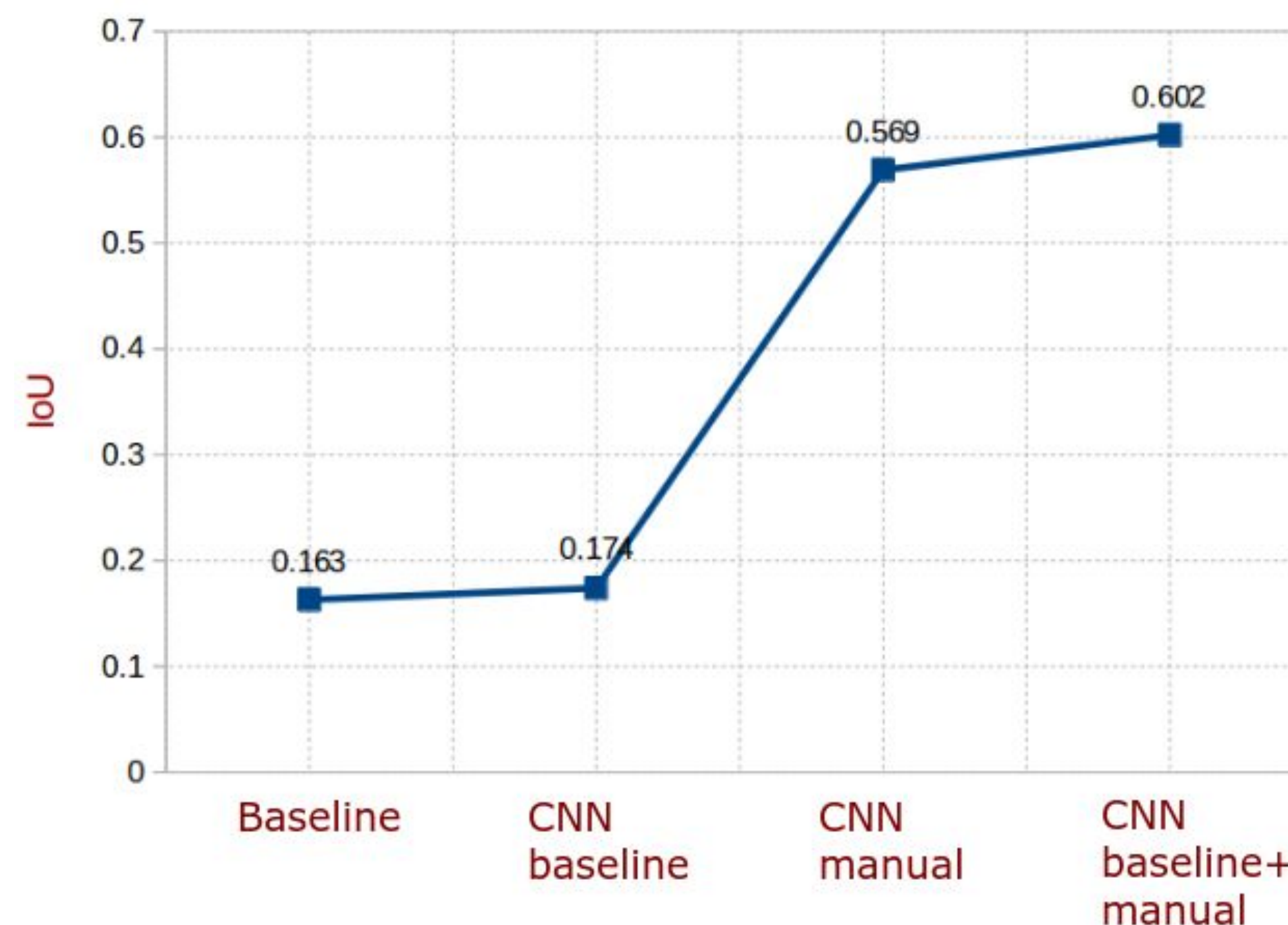
K-means

Mean shift

Filters

A final model was created using only manually extracted data.

Average IoU of 0.569.



*Ciencia para **protegernos***  
*Ciencia para **avanzar***