

BABEŞ BOLYAI UNIVERSITY, CLUJ NAPOCA, ROMÂNIA  
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

# Automatic Detection and Classification of Atmospherical Fronts

– MIRPR report –

**Ploscar Andreea Alina, Muscalagiu Anca Ioana**  
Informatica Engleza, group 935

2022-2023

## Abstract

Automatic determination of fronts from atmospheric data serves as an important aid for weather forecasting, as atmospheric fronts provide strong indicators of certain meteorological characteristics. This paper proposes an application that detects and classifies air fronts in an automatic manner, using Convolutional Neural Networks for pattern recognition in images of type "synoptic map", which contain multiple weather parameters. The user can upload a synoptic map and the application informs him if in this image there is an existing atmospheric front on Romania's territory. If there is any front detected, then the algorithm will also classify it in one of three categories: cold, warm or mixed front. The data used to train and test this CNN model is collected from different meteorological stations around the world, between 2013 to 2020.

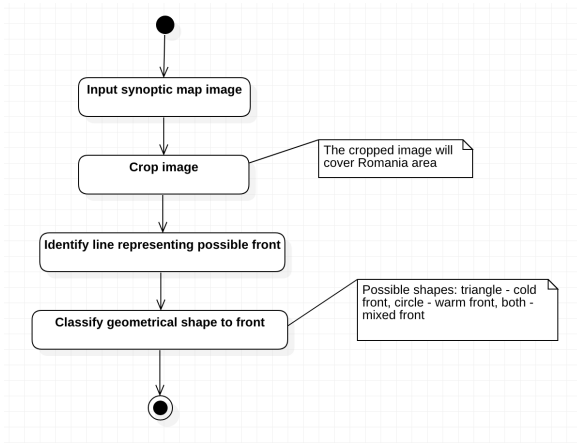


Figure 1: Work Flow

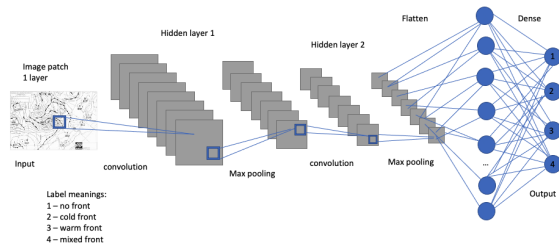


Figure 2: Initial Model

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	What? Why? How? . . . . .	1
1.2	Paper structure and original contribution(s) . . . . .	1
<b>2</b>	<b>Scientific Problem</b>	<b>3</b>
2.1	Problem definition . . . . .	3
<b>3</b>	<b>State of the art - Related work</b>	<b>5</b>
<b>4</b>	<b>Investigated approach</b>	<b>6</b>
<b>5</b>	<b>Application (Study case)</b>	<b>8</b>
5.1	App's description and the main functionalities . . . . .	8
5.2	App's design . . . . .	8
5.3	Implementation . . . . .	8
5.4	Testing of the codebase . . . . .	8
5.5	Numerical validation . . . . .	8
5.5.1	Methodology . . . . .	9
5.5.2	Data . . . . .	10
5.5.3	Results . . . . .	11
5.5.4	Discussion . . . . .	12
<b>6</b>	<b>Conclusion and future work</b>	<b>13</b>

# List of Tables

6.1	The parameters of the PSO algorithm (the micro level algorithm) used to compute the fitness of a GA chromosome. . . . .	14
-----	---	----

# List of Figures

1	Work Flow . . . . .	
2	Initial Model . . . . .	
2.1	Synoptic Map . . . . .	3
2.2	Cold Front . . . . .	4
2.3	Warm Front . . . . .	4
2.4	Stationary Front . . . . .	4
2.5	Occluded Front . . . . .	4
5.1	CNN model . . . . .	9
5.2	Confusion matrix as a Heatmap . . . . .	10
5.3	Accuracy evolution . . . . .	10
5.4	Synoptic Map . . . . .	11
5.5	Overall Accuracy Evolution . . . . .	11
5.6	Accuracy Per Class Evolution . . . . .	11

# List of Algorithms

# Chapter 1

## Introduction

### 1.1 What? Why? How?

This paper aims to use Convolutional Neuronal Networks on synoptic maps collected from meteorological stations to determine the existence and category of fronts over Romania's territory in a specific moment. Front detection is a subject addressed very little in the past years and only for broader territories like Europe and America. Our model proposes to improve the accuracy of air fronts detection for smaller regions like Romania.

Detecting and classifying atmospherical fronts recently become an important topic not only in the meteorological field, but also in the medical field. The specialists have observed a certain correlation between the number of strokes and the particular fronts existent on a certain territory over a longer period of time. Therefore, developing an algorithm that can correctly and automatically determine and classify the type of front over a specific territory is an important step in researching the exact correlation of health risks and specific weather characteristics that persons suffering of a stroke were exposed to over a longer period of time, helping in preventing and reducing the risk of this condition occurring again.

### 1.2 Paper structure and original contribution(s)

The research presented in this paper advances the theory, design, and implementation of several particular models.

The main contribution of this report is to present an intelligent algorithm for solving the problem of determining and classifying atmospherical fronts.

The second contribution of this thesis consists of building an intuitive, easy-to-use and user friendly software application. Our aim is to build an algorithm that will help specialists in the meteorological field forecast different weather characteristics on smaller regions, more specifically on Romania's territory. The model proposed in this report aims to serve as a starting point for further research on the correlation between weather attributes and stroke frequency in certain area.



# Chapter 2

## Scientific Problem

### 2.1 Problem definition

The application detects air fronts represented by means of geometric shapes (lines, semicircles, triangles) in synoptic maps with the help of Convolutional Neural Networks , then classify the fronts in one of 4 classes: no existing air front, warm, cold or mixed front.

The input data of the problem are represented by sets of "synoptic map" type images, collected from different weather stations, in which different meteorological characteristics are represented: air pressure, temperature and humidity, baric tendency, wind direction and speed. Thus, the application aims to detect the location of Romania from the analyzed weather maps, cut the images to contain only the territory of interest and process them to detect the atmospheric sources, represented by different geometric shapes.

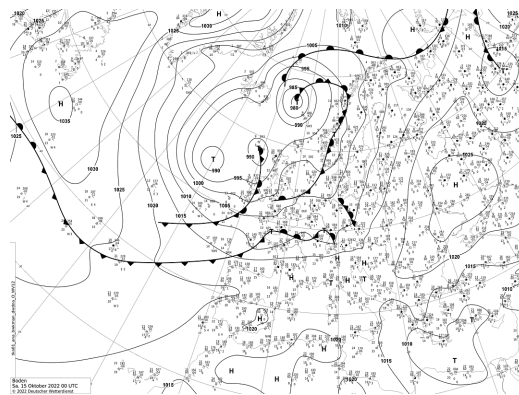


Figure 2.1: Synoptic Map

The atmospheric front (or air front) represents the transition between two air masses different in density or temperature. Their contact can cause radical weather changes, such as precipitation, temperature or pressure variations. In synoptic type weather maps, the two air masses are delimited by continuous lines, and the type of air front is determined by different geometric shapes: semicircle for warm fronts, triangles for cold fronts and alternating triangles and semicircles for mixed fronts. The type of the front is determined by the dominating type of air mass: cold or warm.



Figure 2.2:  
Cold Front



Figure 2.3:  
Warm Front



Figure 2.4:  
Stationary  
Front



Figure 2.5: Oc-  
cluded Front

The output for this model is represented by one of the following classes: No air front, Cold, Warm or Mixed front. For the purpose of this paper both the Stationary and the Occluded fronts are considered as Mixed fronts. The labels used for this classifications are: 1 - no front, 2 - cold front, 3 - warm front, 4 - mixed front.

## Chapter 3

# State of the art - Related work

The detection and classification of air fronts is usually addressed manually, but the demand for an automatic approach increased along with the dataset volume. There have been multiple approaches in creating an automatic model, but the most relevant work for this study is S. Niebler et al.: Automatic detection and classification of fronts, 2022.

This paper uses labeled data from two weather services: the North American National Weather Service and the Deutscher Wetterdienst. The study introduces a deep neural network to detect and classify fronts from multi-level ERA5 reanalysis data, putting an emphasis on simplifying the post processing step. With a supervised learning approach, the method used is a CNN that automatically learns atmospheric features that correspond to the existence of a weather front. For each spatial grid point the algorithm predicts a probability distribution, the likelihood of the point belonging in one of the five classes: warm front, cold front, occlusion, stationary front or background.

In comparison with this paper, our study focuses on one particular area from the eastern Europe, covering Romania. Thus, the output of our algorithm contains a single result, one of the four classes. The occluded and the stationary front mentioned in this paper are considered as mixed fronts in our model because identifying both triangles and circles in our image would lower the precision of classifying both classes, information that is not relevant to us. Another important distinction is the presence of at most one front in our images, because the covered area is smaller, therefore there is no possibility for wrongly classifying mixed front with alternating cold and warm fronts.

## Chapter 4

# Investigated approach

The approach in this paper involves a few steps of preprocessing of the input data. The initial images received from the weather stations cover the Europe continent, containing multiple or even all types of fronts, making the result of the classification meaningless. In order to have the input data which can be classified in one predominant class, the synoptic maps were splitted into 16 equally sized tiles. This method lowers the chance of an image containing multiple fronts, as it covert a smaller area. The resulting images were prelabeled and organized in training set, validation set and test set, each one of them being classified further in one of the 4 output classes.

The algorithm used is based on the classical model for image recognition through supervised learning, making use of Convolution Neural Networks with multiple convolution, pooling and dropout layers. The CNN is organized in multiple 2D convolution layers, taking into consideration the format of the input image (black and white) which is represented with only one slice, average pooling and dropout layers, which help prevent overfitting. The last step of the neural network is the flattening phase, outputting the probability of the input belonging to each of the 4 classes. For the first layer the number of input channels is 1, because the algorithm uses black and white images. After the 2d convolution is applied, the layer performs in addition a batch normalization and a rectified linear unit function (ReLU). All of the convolutional and pooling layers form the sequential layer, which is followed by the flattening phase.

The training phase starts with no accuracy, slightly increasing over the epochs. In each epoch the algorithm trains the model and iterates over batches of images loaded from the train dataset. The classes are predicted using images from the current batch and the output is compared to the actual labels. The loss function used is cross entropy and it backpropagates the loss into the network. This

allows the parameters to be adjusted according to the computed gradients. After the iteration, the learning rate is adjusted, decreasing with the current epoch. The train loss and accuracy are measured using the predictions and the actual labels for all images from each batch. The testing phase evaluates the model on the test dataset and saved the model if a greater accuracy is obtained.

## Chapter 5

# Application (Study case)

### 5.1 App's description and the main functionalities

Application description

Main functionalities and their specification

Add a flow diagram (very useful for the client)

### 5.2 App's design

- use cases
- diagrams (class diagram, sequence diagram, database structure)

### 5.3 Implementation

Give details about app's implementations (language, technologies, frameworks, special settings, etc.)

### 5.4 Testing of the codebase

Asserts, unitTestings, coverage, etc.

### 5.5 Numerical validation

Explain the experimental methodology and the numerical results obtained with your approach and the state of art approache(s).

Try to perform a comparison of several approaches.

Statistical validation of the results.

### 5.5.1 Methodology

From an experimental point of view, we approached this problem using a dense CNN with the following architecture :

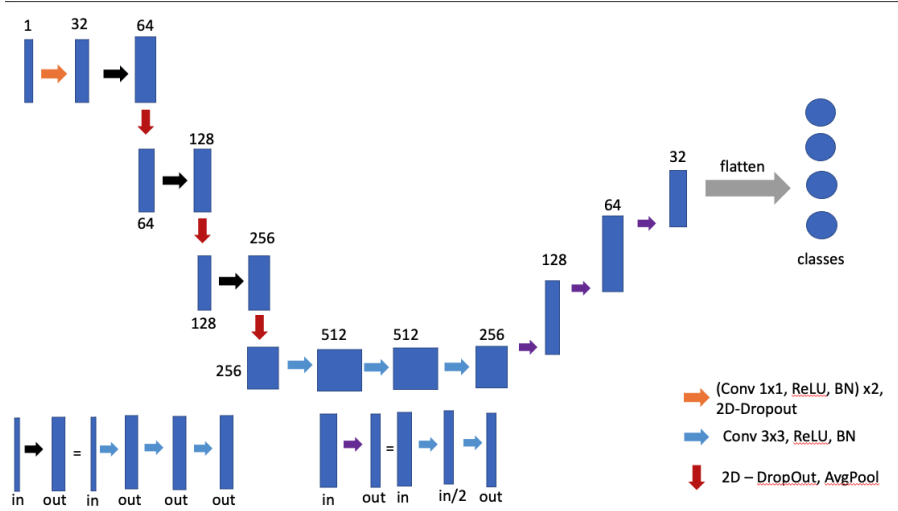


Figure 5.1: CNN model

The structure is inspired by Niebler in [1], being simplified for our use case with front classification in black and white photos. Some of the layers that we removed include the upsampling layers and their following convolution layers.

In order to evaluate our method we are measuring the number of correctly classified fronts, computing the overall and by class accuracy. We also compute a confusion matrix to better visualize the performance of our CNN model. We display this matrix as a heatmap and we plot the evolution with each epoch of the overall accuracy.

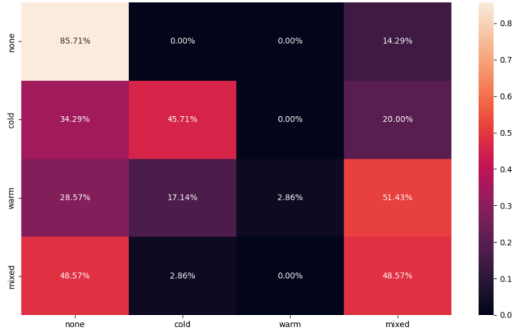


Figure 5.2: Confusion matrix as a Heatmap

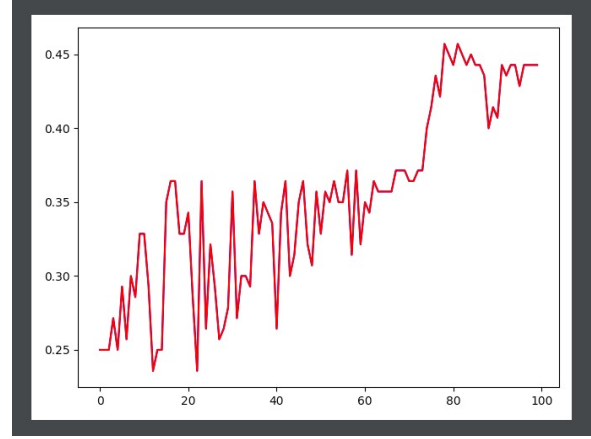


Figure 5.3: Accuracy evolution

This paper tests the hypothesis that atmospheric fronts covering a small area ( less that 350000 sq km) can be classified by a CNN with an accuracy over 80%. Our experiments starts with loading the data, being split in train and validation data. These two are shuffled into batches of 32 images, in order to maximize the probability of having each type of front in all batches to aquire knowledge about all fronts after processing each batch. For each epoch we measure the train loss and train and test accuracy. If the test accuracy is better the best accuracy measured so far we save the model and upgrade the best one. We train the model over 50 epochs, saving intermediary best models.

In comparison with other articles, because the aim of our experiment was classification and not detection of the fronts [1], we were able to measure the performance of our algorithm numerically, as described above, through the number of correctly classified pictures, making use of all the classic performance assessment methods used in the ML field.

To avoid overfitting to one class, we have used equally sized datasets from each category for the training and testing data. In order to increase the performance of our CNN in real life situations we have used both simple, easily identifiable fronts and more complex pictures illustrating multiple types of fronts, having a predominant one. This makes our datasets more realistic and relevant.

### 5.5.2 Data

The consistent datasets were downloaded from Wetterzentrale, a German weather service which provides synoptic maps daily. The format of the map is presented in the picture below:





### 5.5.3 Results

The graph illustrates the average number of nodes in the largest component of a network over 50 iterations. The y-axis represents the average number of nodes, ranging from 0.25 to 0.45. The x-axis represents the iteration number, ranging from 0 to 50. The data shows a general upward trend with significant fluctuations. The average number of nodes starts at 0.25 at iteration 0, rises to about 0.32 by iteration 10, and then continues to fluctuate between 0.28 and 0.38 until iteration 40. After iteration 40, there is a sharp increase to approximately 0.46 by iteration 50.

Iteration	Average Number of Nodes
0	0.25
1	0.25
2	0.28
3	0.30
4	0.30
5	0.32
6	0.31
7	0.32
8	0.28
9	0.28
10	0.31
11	0.29
12	0.30
13	0.33
14	0.31
15	0.28
16	0.28
17	0.30
18	0.30
19	0.35
20	0.34
21	0.36
22	0.35
23	0.38
24	0.34
25	0.38
26	0.28
27	0.35
28	0.25
29	0.24
30	0.36
31	0.34
32	0.35
33	0.27
34	0.29
35	0.34
36	0.34
37	0.38
38	0.32
39	0.28
40	0.26
41	0.33
42	0.35
43	0.35
44	0.35
45	0.35
46	0.34
47	0.34
48	0.46

The graph displays the probability of four weather conditions over 50 time steps. The 'none' condition (orange) is the most frequent, often reaching a probability of 1.0. The 'cold' condition (blue) shows more variation, with peaks around 0.5. The 'warm' condition (red) is the least frequent, staying near 0.0. The 'mixed' condition (green) also fluctuates significantly, often reaching 1.0.

11

#### 5.5.4 Discussion

- Is your hypothesis supported?
- What conclusions do the results support about the strengths and weaknesses of your method compared to other methods?
- How can the results be explained in terms of the underlying properties of the algorithm and/or the data.

## Chapter 6

# Conclusion and future work

Try to emphasise the strengths and the weaknesses of your approach. What are the major shortcomings of your current method? For each shortcoming, propose additions or enhancements that would help overcome it.

Briefly summarize the important results and conclusions presented in the paper.

- What are the most important points illustrated by your work?
- How will your results improve future research and applications in the area?

Item example:

- content of item1
- content of item2
- content of item3

Figure example

... (see Figure ??)

Table example: (see Table 6.1)

Table 6.1: The parameters of the PSO algorithm (the micro level algorithm) used to compute the fitness of a GA chromosome.

Parameter	Value
Number of generations	50
Number of function evaluations/generation	10
Number of dimensions of the function to be optimized	5
Learning factor $c_1$	2
Learning factor $c_2$	1.8
Inertia weight	$0.5 + \frac{rand()}{2}$

# Bibliography

- [1] Stefan Niebler, Annette Miltenberger, Bertil Schmidt, and Peter Spichtinger. Automated detection and classification of synoptic-scale fronts from atmospheric data grids. *Weather and Climate Dynamics*, 3(1):113–137, 2022.