

MIRPR Front Detection

Gheorghe Mara, Bentea Bogdan, Bogdan Iulian, Bodea Stefan

Abstract

Acest proiect are scopul de a cauta fronturile in imaginile meteo.

1 Prima incercare

Pentru inceput am incercat sa extragem doar fronturile din imagine pentru a simplifica munca clasificatorului. Pentru extragere am folosit biblioteci din cv2 care sa aplice un kernel de diferite dimensiuni pe imagine.

Pentru kernelul $K_{1 \times 1} = [1]$ a rezultat imaginea:

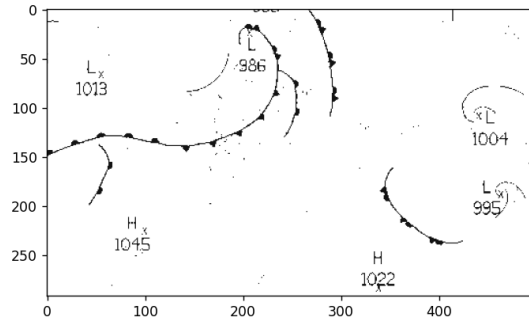


Figure 1: Imaginea rezultata dupa aplicarea kernelului 1×1

Iar pentru kernelul $K_{2 \times 2} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$ am obtinut:

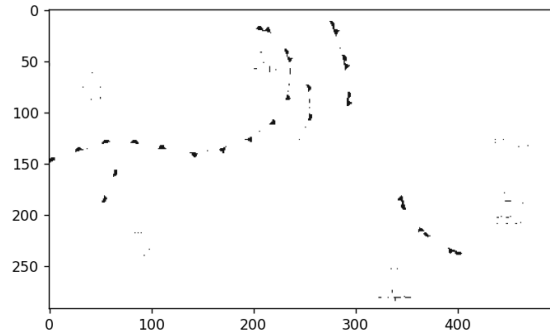


Figure 2: Imaginea rezultata dupa aplicarea kernelului 2×2

2 Continuarea

2.1 Izolare linii

Pentru a obtine o imagine cat mai "curata", in care sa fie vizibile fronturile, am filtrat imaginile in 2 moduri, cele colore si cele alb negru.

Pentru cele colore am creat 3 masti, pentru fiecare culoare a fronturilor. O data aplicate mastile, se obtine o imagine cu fundal negru si fronturile colorate.

In final am transformat imaginea astfel incat sa obtinem fundal alb si fronturi de culoare neagra. Pentru imaginile alb-negru, am incercat sa eliminam acei pixeli negri care au o intensitate mica.

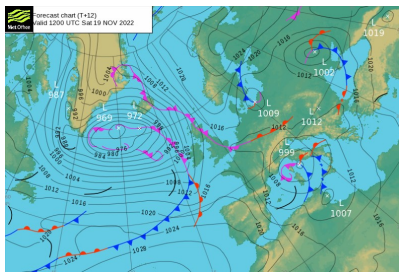


Figure 3: Map 1

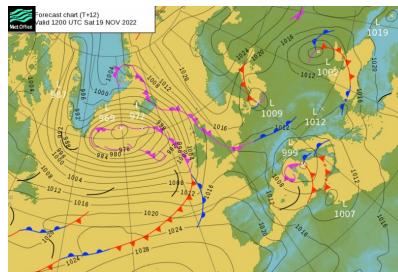


Figure 4: Map 2

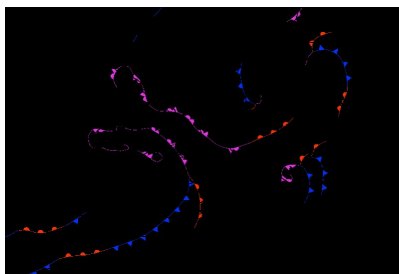


Figure 5: Map 3

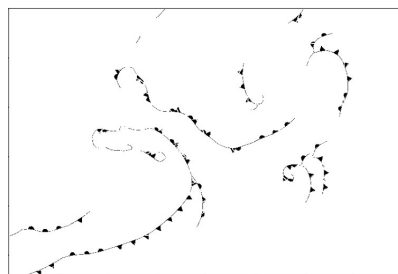


Figure 6: Map 4

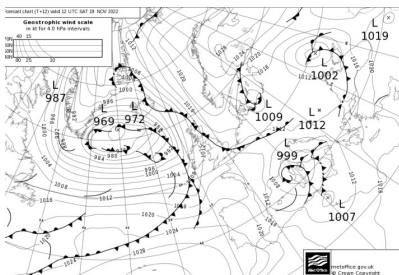


Figure 7: Map 5

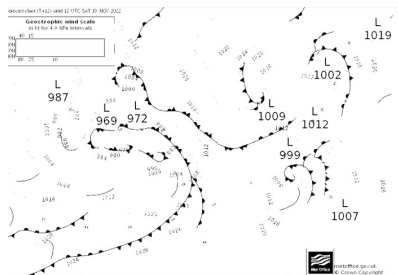


Figure 8: Map 6

2.2 Creare si antrenare model

Am creat un model de tipul Sequential cu urmatoarele layere:

1. Rescaling, pentru a normaliza pixeli care primeste ca si input o poza de dimensiunile (50, 50, 3)
2. Un layer de convolutie cu 16 kerneluri urmat de MaxPooling.
3. Un layer de convolutie cu 32 de kerneluri si MaxPooling.
4. Un layer de convolutie cu 64 de kerneluri si MaxPooling.

5. Un layer de Flatten dupa care incepe partea de componenta conexa completa care are pe primul strat 128 de noduri.
6. Un layer de 3 noduri reprezentand nodurile de iesire.

Pentru a compila modelul am folosit optimizer-ul 'adam' si metrica de acuratete.

Am antrenat modelul pentru a putea recunoaste fronturile cold, warm si occluded. A primit ca si date de antrenament crop-uri din imaginile reale ale acestor date.

Ca si acuratete a ajuns la 0.7 pentru datele de validare. Dupa care parcurgem imaginea filtrata si verificam fiecare patrat de dimensiunile (50, 50, 3) pe care-l clasificam cu modelul, daca modelul il clasifica "sigur" ca un anume tip de front atunci acel patrat este considerat un anume tip de front.

Model: "sequential_7"

Layer (type)	Output Shape	Param #
rescaling_7 (Rescaling)	(None, 50, 50, 3)	0
conv2d_21 (Conv2D)	(None, 50, 50, 16)	448
max_pooling2d_21 (MaxPooling2D)	(None, 25, 25, 16)	0
conv2d_22 (Conv2D)	(None, 25, 25, 32)	4640
max_pooling2d_22 (MaxPooling2D)	(None, 12, 12, 32)	0
conv2d_23 (Conv2D)	(None, 12, 12, 64)	18496
max_pooling2d_23 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten_7 (Flatten)	(None, 2304)	0
dense_14 (Dense)	(None, 128)	295040
dense_15 (Dense)	(None, 3)	387

=====
 Total params: 319,011
 Trainable params: 319,011
 Non-trainable params: 0

Figure 9: Model de tip Sequential

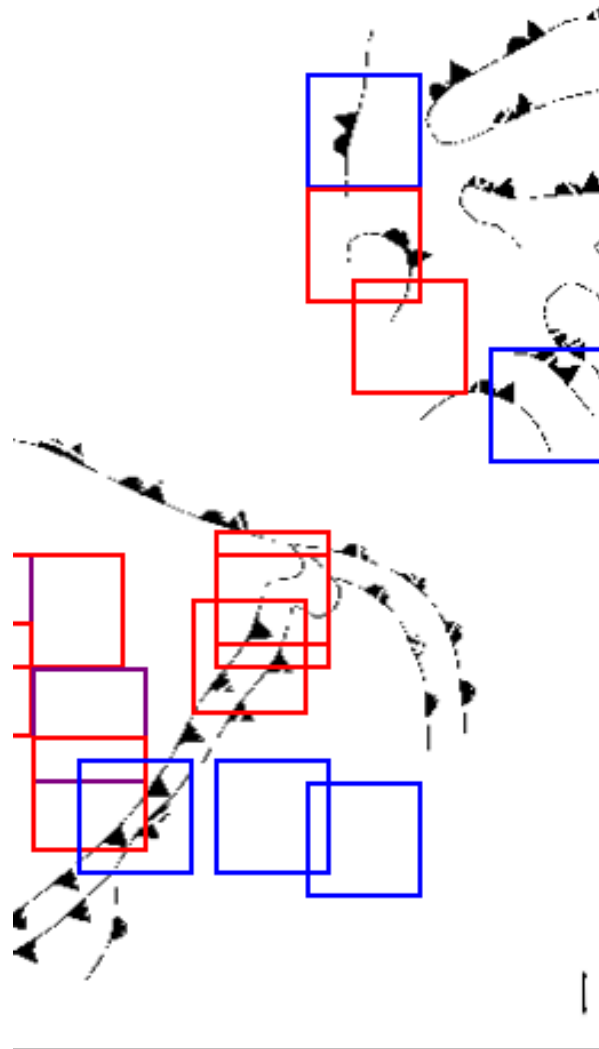


Figure 10: Rezultat 1



Figure 11: Rezultat 2

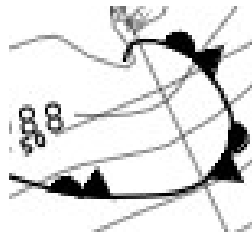


Figure 12: Rezultat 3



Figure 13: Rezultat 4

3 Imbunatatirea algoritmului

3.1 Metodologie

Folosim un model pre-antrenat, numit VGG-16. Acest model a fost propus de Karen Simonyan si Andrew Zisserman, membri ai "Visual Geometry Group" din cadrul Universitatii Oxford. VGG-16 atinge acuratete de pana la 92.7% pe setul de date ImageNet (ce contine peste 14 milioane de imagini, ce apartin de 1000 de clase).

In cazul algoritmului nostru, am adaugat trei layere fully connected, unul cu 1024 de noduri, urmatorul cu 512 si ultimul cu 4 noduri, egal cu numarul de clase. Am modificat modelul VGG-16 pentru a clasifica 4 clase pe care le gasim in pozele cu fronturile, adica clasa "COLD" care este reprezentata de simbolul triunghi, clasa "WARM" care este reprezentata de simbolul semicerc, clasa "OCCLUDED" care este reprezentata de cele doua simboluri alaturate si clasa "WHITE" care reprezinta tot ce nu este una din cele trei clase de interes.

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 1, 1, 512)	14714688
flatten_1 (Flatten)	(None, 512)	0
dense_3 (Dense)	(None, 1024)	525312
dropout_2 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 512)	524800
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 4)	2052

```
=====  
Total params: 15,766,852  
Trainable params: 1,052,164  
Non-trainable params: 14,714,688  
+ Code + Markd
```

Figure 14: Model de tip VGG-16

Pentru a obtine o clasificare mai buna, am decis sa impartim imaginea filtrata in zone de dimensiune mai mica, adica imagini de 32x32 px, fata de la prima versiune unde imaginile aveau dimensiunea de 100x100 px. O imagine acum contine un singur simbol (Figurile 15 - 23), nu o linie cu un sir de simboluri.



Figure 15: Cold symbol



Figure 16: Cold symbol



Figure 17: Cold symbol



Figure 18: Warm symbol



Figure 19: Warm symbol



Figure 20: Warm symbol



Figure 21: Occluded symbol



Figure 22: Occluded symbol



Figure 23: Occluded symbol

Am avut doua iteratii ale modelului inainte de aceasta unde am folosit, o data imaginile mai mari in varianta color si in varianta alb-negru. In variantele anterioare, imaginile nu contineau un singur simbol ci mai multe de acelasi tip. (Figurile 24 - 29).

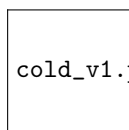


Figure 24: Cold b&w

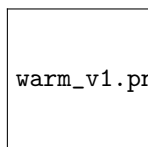


Figure 25: Warm b&w

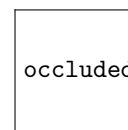


Figure 26: Occluded b&w

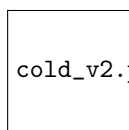


Figure 27: Cold color

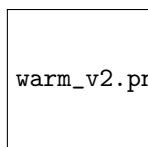


Figure 28: Warm color

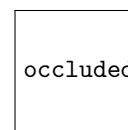


Figure 29: Occluded color

3.2 Analiza comparative

	Varianta 1	Varianta 2	Varianta 3
Acuratete	0.7251	0.6571	0.8750
Timpi de inferenta (ms)	3511.568115	3349.47949	2775.35156
Matricea de confuzie	Figura 30	Figura 31	Figura 32
Experiment	Figura 33	Figura 34	Figura 35

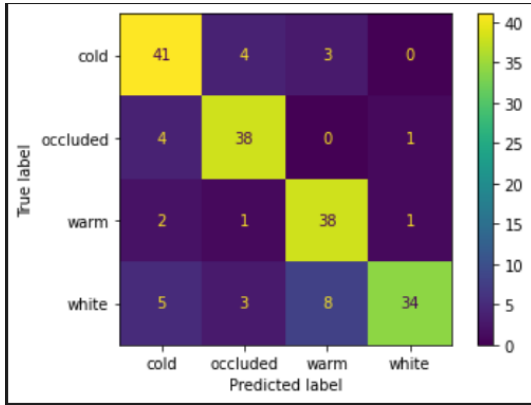


Figure 30: Matricea de confuzie pentru varianta 1

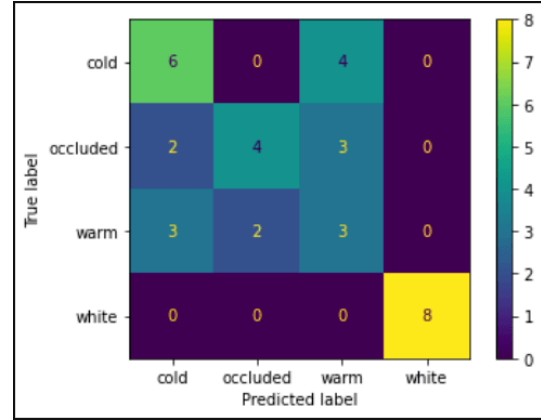


Figure 31: Matricea de confuzie pentru varianta 2

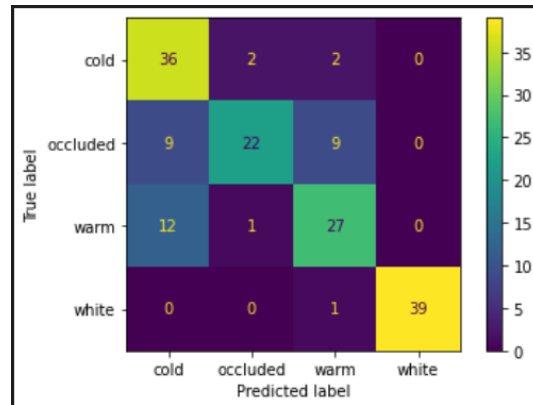


Figure 32: Matricea de confuzie pentru varianta 3

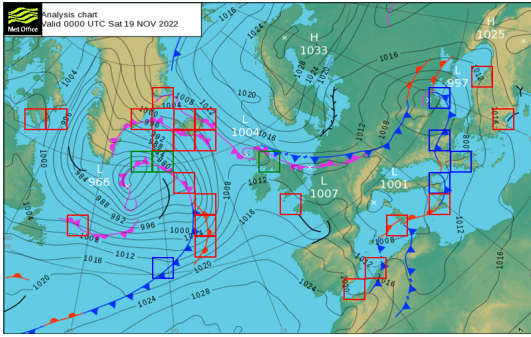


Figure 33: Experiment varianta 1

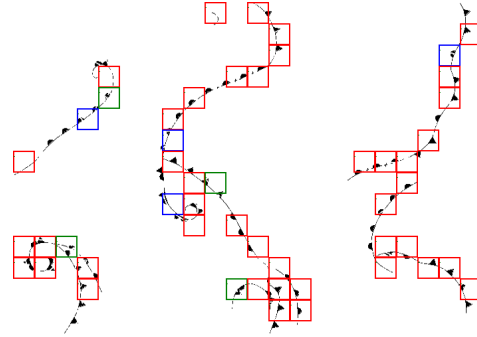


Figure 34: Experiment varianta 2

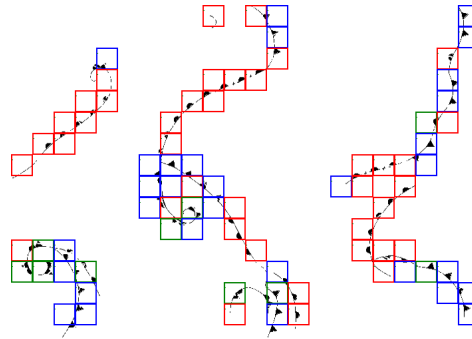


Figure 35: Experiment varianta 3

3.3 Rezultate

Metrica	Cold	Warm	Occluded	White
Precizia	0.54	0.64	0.80	0.97
Rapelul	0.93	0.45	0.50	0.97

Pentru versiunea 3 acuratetea obtinuta este aproximativ 87% pe datele de validare iar pe un exemplu real modelul se descurca cam asa.

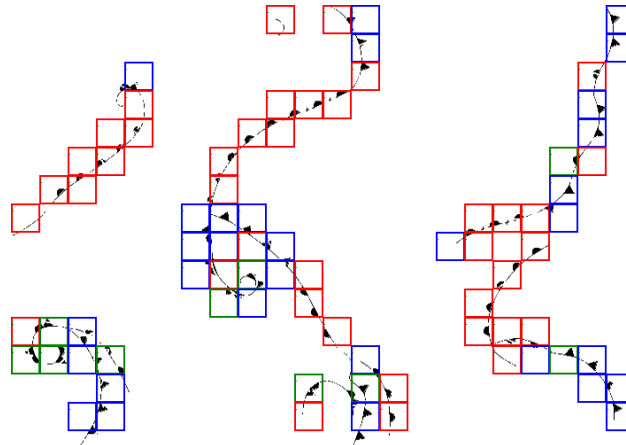


Figure 36: Experiment varianta 3