

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST  
Obor č. 18: Informatika

Frontanim – otevřený  
systém pro predikci polohy  
atmosférických front

Pavel Pernička  
Zlínský kraj

Zlín 2024

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST  
Obor č. 18: Informatika

# Frontanim – otevřený systém pro predikci polohy atmosférických front

## Frontanim – opensource atmospheric front position prediction system

**Autoři:** Pavel Pernička

**Škola:** Gymnázium a Jazyková škola s právem státní jazykové zkoušky  
Zlín, nám. T. G Masaryka 2734, 760 01 Zlín

**Kraj:** Zlínský kraj

**Konzultant:** Mgr. Radek Janošík, Ph.D.

Zlín 2024

# Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

Ve Zlíně dne 17. 3. 2024 Pavel Pernička

# Poděkování

Rád bych v první řadě poděkoval svému konzultantovi, panu Mgr. Radku Janoščíkovi, Ph.D., za pravidelné schůzky, během kterých jsme diskutovali postup práce, a za domluvení možnosti využití výpočetních kapacit na Katedře informatiky Univerzity Palackého v Olomouci pro trénování neuronové sítě.

Dále bych chtěl vyjádřit své poděkování panu Ing. Davidu Šaurovi, Ph.D. z Univerzity Tomáše Bati ve Zlíně za jeho cenné rady týkající se meteorologických aspektů této práce. Také nemůžu nezmínit našeho učitele semináře z informatiky, pana Mgr. Michala Mikláše, bez jehož povzbuzování bych se na Středoškolskou odbornou činnost nedal.

# Anotace

Předmětem této práce je návrh systému pro predikci polohy a typu atmosférických front v dané oblasti na základě výstupních dat numerických modelů pro předpověď počasí. Systém kombinuje využití datových produktů založených na fyzikálních parametrech s generativní neuronovou sítí. Součástí řešení je otevřený zdrojový kód.

# Klíčová slova

atmosférická fronta, počasí, neuronová síť, NWP model, synoptická meteorologie

# Annotation

The aim of this work is the design of a system for predicting the position and type of atmospheric fronts in a given area, based on the output data from numerical weather prediction models. The system combines the use of data products based on physical parameters with a generative neural network. Solution includes opensource code.

# Keywords

atmospheric front, weather, neural network, NWP model, synoptic meteorology

# Obsah

<b>1</b>	<b>Úvod</b>	<b>6</b>
<b>2</b>	<b>Meteorologická teorie</b>	<b>7</b>
2.1	Atmosférická fronta . . . . .	7
2.1.1	Studená fronta . . . . .	8
2.1.2	Teplá fronta . . . . .	8
2.1.3	Okluzní fronta . . . . .	8
2.2	Ukazatele výskytu front . . . . .	9
2.2.1	Horizontální teplotní gradient . . . . .	9
2.2.2	Termální frontální parametr . . . . .	10
2.2.3	Rychlost proudění větru ve výšce . . . . .	10
2.2.4	Frontogenetická funkce . . . . .	11
2.2.5	Q-vektory . . . . .	13
2.2.6	Relativní vorticity . . . . .	14
2.2.7	Prvky pro subjektivní analýzu . . . . .	16
<b>3</b>	<b>Použité technologie</b>	<b>17</b>
3.1	Python 3 . . . . .	17
3.2	PyTorch . . . . .	17
3.3	OpenCV . . . . .	18
3.4	GDAL lib . . . . .	18
3.5	Skript shellu . . . . .	19
3.6	GTK 3 . . . . .	20
3.7	Git . . . . .	20
<b>4</b>	<b>Architektura softwaru</b>	<b>21</b>
4.1	Vstup meteorologických dat . . . . .	21
4.2	Vstup polohy front . . . . .	22
4.3	Zpracování . . . . .	22
4.4	Výstup . . . . .	23
<b>5</b>	<b>Zpracování obrazových dat</b>	<b>24</b>
5.1	Extrakce front . . . . .	24
5.2	Georeference . . . . .	26
5.3	Strojově čitelný formát . . . . .	26
<b>6</b>	<b>Zpracování dat z NWP modelů</b>	<b>28</b>
6.1	Práce s daty . . . . .	28
6.2	Složení produktu . . . . .	29
6.3	Konverze na obrázek . . . . .	29
6.4	Výběr validních dat . . . . .	30
<b>7</b>	<b>Neuronová síť</b>	<b>31</b>
7.1	Pix2pix . . . . .	31
7.1.1	Generátor . . . . .	32
7.1.2	Diskriminátor . . . . .	32
7.2	Proces tréninku . . . . .	33

7.3	Ztrátová funkce . . . . .	34
7.4	Destrukce kvality . . . . .	35
7.5	Chybějící fronty . . . . .	36
<b>8</b>	<b>Úspěšnost</b>	<b>37</b>
8.1	Metodika . . . . .	37
8.2	Výsledky . . . . .	38
<b>9</b>	<b>Závěr</b>	<b>39</b>
9.1	Budoucí rozvoj . . . . .	39
9.1.1	Data . . . . .	39
9.1.2	Zdokonalení numerických produktů . . . . .	39
9.1.3	Webová aplikace . . . . .	39
<b>10</b>	<b>Bibliografie</b>	<b>40</b>
	<b>Seznam obrázků</b>	<b>43</b>
	<b>Seznam ukázek kódu</b>	<b>43</b>
	<b>Seznam tabulek</b>	<b>43</b>
	<b>Přílohy</b>	<b>44</b>

# 1 Úvod

Tento dokument popisuje koncept systému pro vytvoření grafického znázornění synoptické situace ve formě frontálních linií, které je používáno jak odborníky, tak amatérskou veřejností. V současné praxi jsou tyto synoptické mapy zakreslovány pomocí uzavřených systémů, případně ručně meteorology, proto je mým cílem vytvořit volně dostupný systém, který navíc používá méně tradiční způsob – použití neuronové sítě.

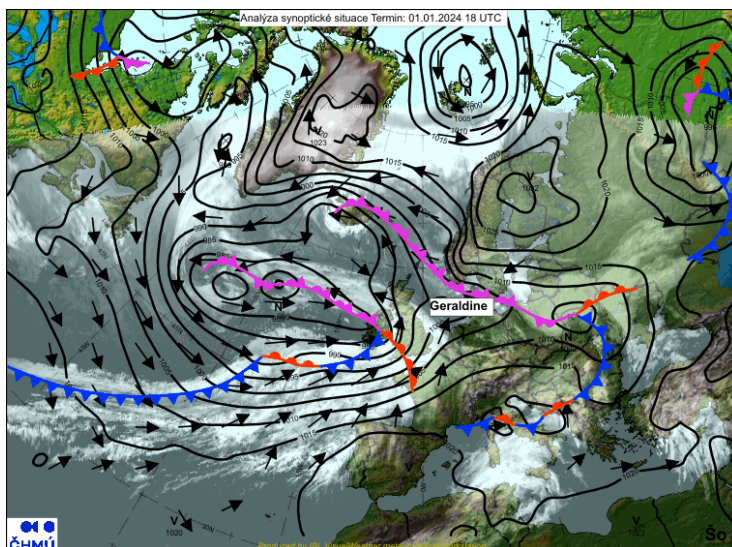
Hlavní inspirací pro tento projekt byl nápad ve studentském meteorologickém klubu na *Gymnáziu a Jazykové škole s právem státní jazykové zkoušky Zlín* ukazovat ve spotech s předpovědí počasí také krátkou animaci, jak se budou pohybovat fronty v následujících dnech. Jelikož by to kvůli absenci otevřeného softwaru k tomuto účelu bylo velmi časově náročné, rozhodl jsem se této problematice věnovat tuto práci, která je celou dobu vyvíjena s otevřeným zdrojovým kódem pro snadnější budoucí vývoj.

Aby byl projekt alespoň z části použitelný i mimo svůj hlavní účel, je rozdělen na několik dílčích skriptů, které se zároveň chovají jako utility pro příkazovou řádku. Typickým příkladem využití této vlastnosti je georeference a převod obrázku analýzy synoptické situace z meteorologického institutu do formátu *GeoJSON*, který umožňuje využití v geografických informačních systémech a dalších aplikacích. Obdobně univerzální využití by měl mít i generátor mezi-produktů z numerického předpovědního modelu a další z vytvořených utilit.



## 2 Meteorologická teorie

Pro bližší pochopení principu fungování zakreslování frontálních linií je potřeba znát základní pojmy ze synoptické meteorologie. Tento obor se zabývá výzkumem velkoprostorových atmosférických jevů, jejich analýzou a předpovědí. Mezi tyto jevy se počítají tlakové útvary (cyklony a anticyklony), vzduchové hmoty a atmosférické fronty.[1] K zaznamenání plošného výskytu těchto jevů se používají synoptické mapy, kde je použito schematické označení ve formě symbolů a linií. Ukázková mapa synoptické situace je znázorněna na obrázku 1.



Obrázek 1: Mapa analýzy synoptické situace z 1. 1. 2024 12:00 UTC[10]

### 2.1 Atmosférická fronta

Nerovnoměrné ohřívání zemského povrchu způsobené různými faktory vede také k nerovnoměrnému ohřívání atmosféry. V důsledku toho se v různých regionech atmosféra liší svými vlastnostmi, z nichž nejdůležitější je teplota vzduchu, s ní související hustota a obsah vodních par. Tyto oblasti se stejnými parametry nazýváme vzduchové hmoty.

Přechodové zóny mezi vzduchovými hmotami se vyznačují prudkými změnami meteorologických prvků, zatímco vnitřní oblasti vzduchových hmot jsou díky setrvačnosti charakteristické pomalými a pozvolnými změnami. [2]

V těchto přechodových zónách, pokud jsou dostatečně výrazné, se vyskytují atmosférické fronty, které značí charakteristiky rozhraní mezi vzduchovými hmotami. Je důležité zdůraznit, že fronta není pouze linie na zemském povrchu, ale prostorová zóna, která je kvůli rozdílným hustotám studeného a teplého vzduchu skloněna k zemi. <sup>1</sup> [3] Z tohoto vyplývá i vliv fronty na počasí, který se

<sup>1</sup>Nemusí se s ní však protnout. Někdy vzduchové hmoty přes území mohou přejít i bez toho.

projevuje v širším pásu území. Tyto projevy se liší v závislosti na vzdálenosti od rozhraní a jeho typu.

### **2.1.1 Studená fronta**

Jedním ze základních typů frontálního rozhraní je studená fronta. Je jednou z nejlépe pozorovatelných, protože přináší velké množství vertikálně rozvinuté oblačnosti, často doprovázené dešťovými přeháňkami a v letním období také bouřkami, které bývají v případě výrazného teplotního rozdílu mezi vzduchovými masami často uspořádané do linií.

Z hlediska stavby jde o rozhraní mezi rychleji se pohybující chladnější vzduchovou hmotou, která se přemísťuje na stranu pomaleji se pohybující teplejší vzduchové hmoty.[4] Pohyb je valivý, rozhraní dostává tvar tupého klínu studeného vzduchu podsouvajícího se pod teplý vzduch. Ten se vlivem vytlačení do vyšších vrstev troposféry ochlazuje a kondenzuje, což má za důsledek vznik oblačnosti.

### **2.1.2 Teplá fronta**

Jedná se o teplou vzduchovou hmotu, která se pohybuje směrem ke studené vzduchové hmotě a nasouvá se na ní. Postupným výstupem teplého vzduchu vzniká charakteristický rozsáhlý oblačný systém doprovázený stratiformními srážkami, jejichž pás může být široký až 400 km.[6]

Přechod této fronty se vyznačuje charakteristickou sekvencí oblačných druhů. Nejprve můžeme sledovat řasovitá oblaka v nejvyšších oblačných patrech, které lze vzhledem k pomalému pohybu fronty sledovat s dlouhým časovým předstihem. Tato oblaka následně vystřídají altocumuly a altostraty ze středních oblačných pater. V konečné fázi se na obloze objevují atratocumuly a nimbostraty v nejnižších vrstvách, které bývají doprovázeny dešťovými srážkami.[5]

### **2.1.3 Okluzní fronta**

Vzhledem k rychlejšímu pohybu studených front ve srovnání s teplejšími může dojít k bodu, kdy se oba typy front spojí. Linie tohoto spojení se nazývá okluzní fronta. Dochází zde ke spojení obou přízemních sektorů studeného vzduchu, které společně vytlačují teplý vzduch nad sebe.[5]

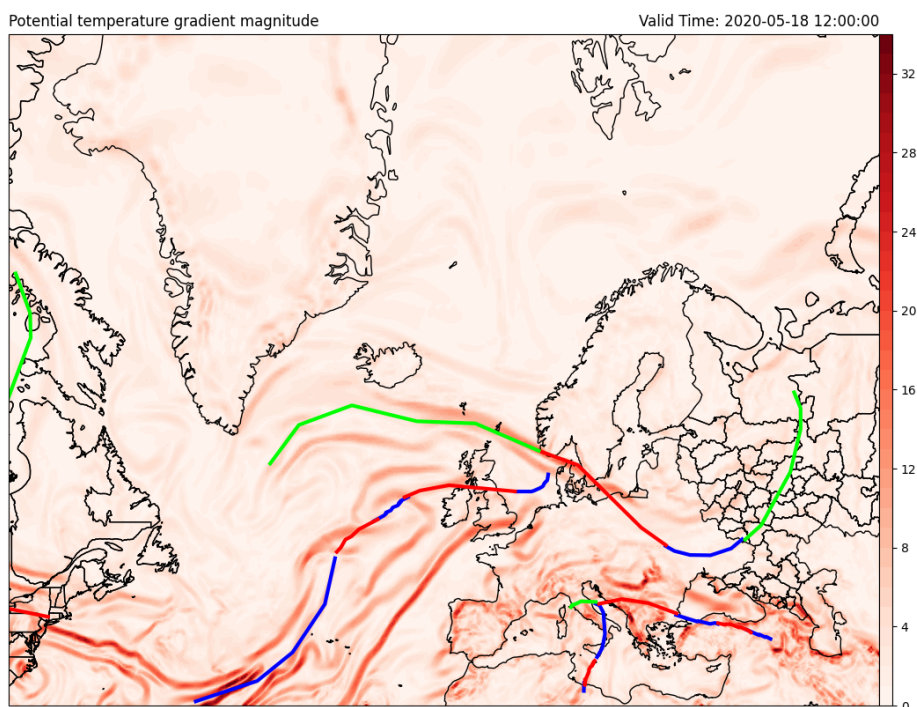
Charakteristika počasí v této situaci je poněkud komplikovanější, jelikož kombinuje projevy obou předchozích front. Proto okluzní frontu dělíme na teplou a studenou. Pokud je vzduchová hmota před okluzní frontou teplejší než ta za ní, jedná se o studenou okluzní frontu, kde většinou převládá charakter studené fronty. V opačném případě jde o teplou okluzi.[7]

## 2.2 Ukazatele výskytu front

Již od počátku používání výpočetní techniky v meteorologii existuje snaha o objektivní znázornění atmosférických front na mapách. Tradiční metody analýzy front se spoléhají na zkušenost a subjektivní úsudek meteorologa, což je časově náročný proces a může vést k nekonzistenci. Objektivní metody analýzy front se snaží tyto problémy eliminovat automatickým rozpoznáním front na základě numerických dat.

### 2.2.1 Horizontální teplotní gradient

Jak již vyplývá z předešlé definice, atmosférická fronta leží na rozhraní rozdílných vzduchových mas, což také znamená, že se bude projevovat zvětšováním horizontálních gradientů vlastností těchto hmot[8]. To je znázorněno na obrázku 2, kde vidíme grafické vyjádření velikosti teplotního gradientu ve výšce 850 hPa. Je zřejmé, že jen tento předpoklad na určení polohy fronty nebude dostačující.



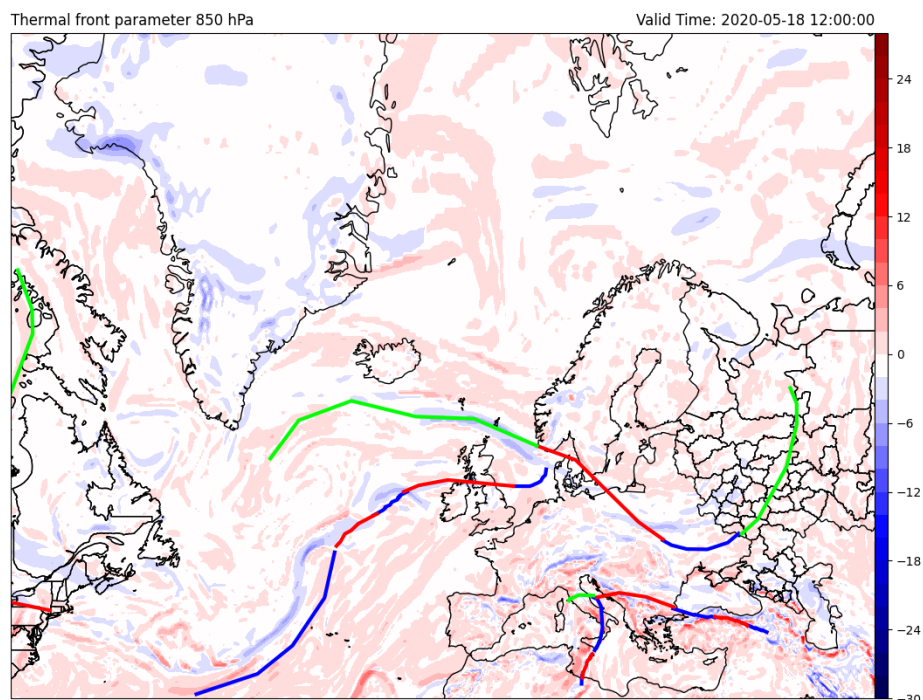
Obrázek 2: Velikost gradientu potenciální teploty

### 2.2.2 Termální frontální parametr

Vhodnější alternativou pro objektivní analýzu synoptické situace může být termální frontální parametr (TFP), který z teplotního gradientu vychází. Matematicky jej formulovali v 60. letech ve své publikaci *EXPERIMENTS IN NUMERICAL OBJECTIVE FRONTAL ANALYSIS* pánové Renard & Clarke[14]. Definován následujícím předpisem:

$$TFP = -\nabla|\nabla\theta_e| \cdot \frac{\nabla\theta_e}{|\nabla\theta_e|}$$

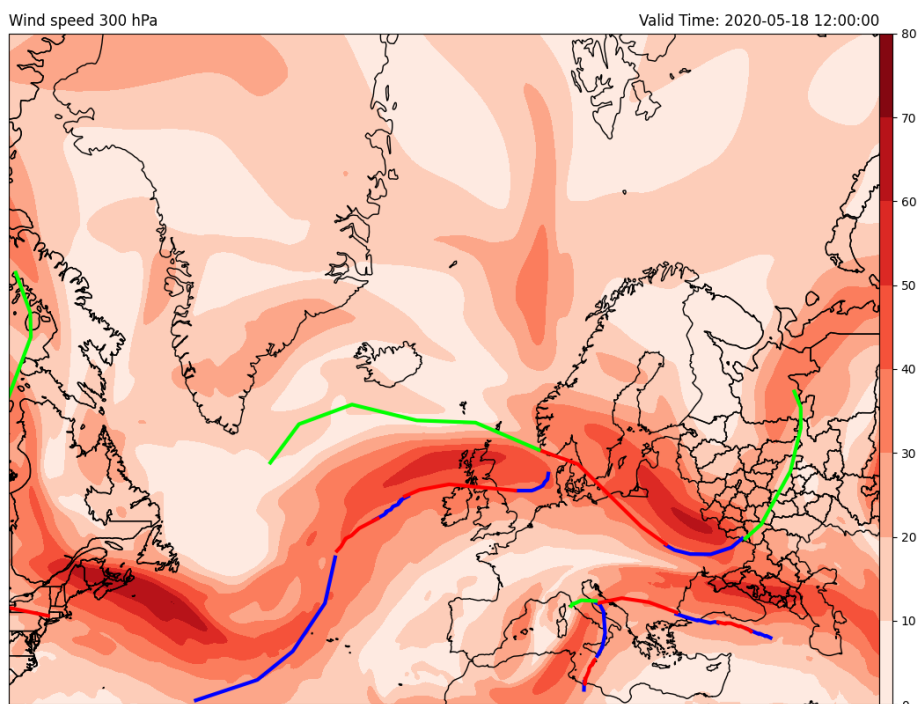
Kde první činitel vyjadřuje změnu teplotního gradientu  $\nabla\theta_e$ , druhý činitel pak projekci této změny do směru teplotního gradientu[9]. Pokud aplikujeme vzorec na modelová data, získáme obrázek 3 ukazující trochu lepší přehled.



Obrázek 3: Termální frontální parametr v hladině 850 hPa

### 2.2.3 Rychlost proudění větru ve výšce

Jelikož je prostředí mezi vzduchovými hmotami velmi dynamické, jako další ukazatel může sloužit vítr. Vzhledem k prostorové rozsáhlosti použijeme hladinu 300 hPa. Ta nám mimo jiné díky tomu, že není příliš ovlivňována zemským povrchem, poskytne relevantní informaci, jestli se na místech zvýšené rychlosti udávají výrazné změny v meteorologických prvcích. V porovnání s pozicemi front na obrázku 4 vidíme korelaci.



Obrázek 4: Rychlost proudění větru v hladině 300 hPa

#### 2.2.4 Frontogenetická funkce

Pokročilejší metodou objektivní frontální analýzy je frontogenetická funkce, kterou formuloval *Howard B. Bluestein* v publikaci *Synoptic-dynamic Meteorology in Midlatitudes: Observations and theory of weather systems*. [11] Zde autor odvozuje vzorec pro výpočet frontogeneze ve dvourozměrném prostoru ( $F$ ) pomocí potenciální teploty ( $\theta$ ), divergence ( $\delta$ ), deformace ( $D$ ) a úhlu mezi izotermami a osou dilatace teplého vzduchu ( $\beta$ ):

$$F = \frac{1}{2} |\nabla_p \theta| [D \cos(2\beta) - \delta]$$

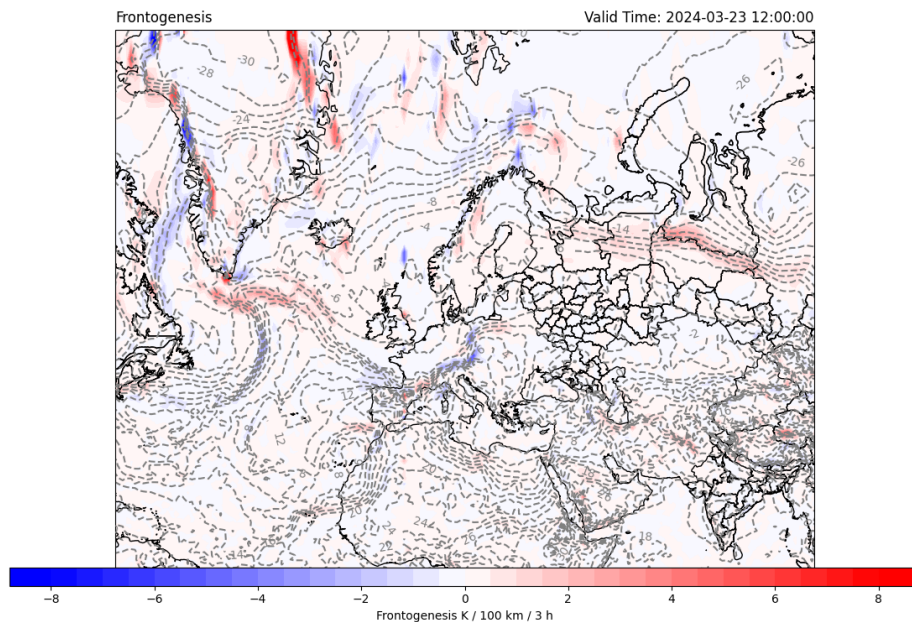
Tento vzorec je implementován v Python knihovně *MetPy* [13], což umožňuje výrazně snadnější výpočet z dat z numerického modelu, jelikož implementace obsahuje výpočet dílčích veličin, tudíž je jako vstup potřeba jen potenciální teplota a vektor větru. Výsledek s daty z modelu GFS je zobrazen na obrázku 5.

Když tento výstup porovnáme s předpovědní synoptickou mapou na stejné datum (obrázek 6) a zanedbáme odchylku vzniklou kvůli různým zdrojům dat, tak vidíme, že frontogeneze pasuje na výrazné fronty. Zejména si pak můžeme povšimnout výrazné studené fronty nad střední Evropou.

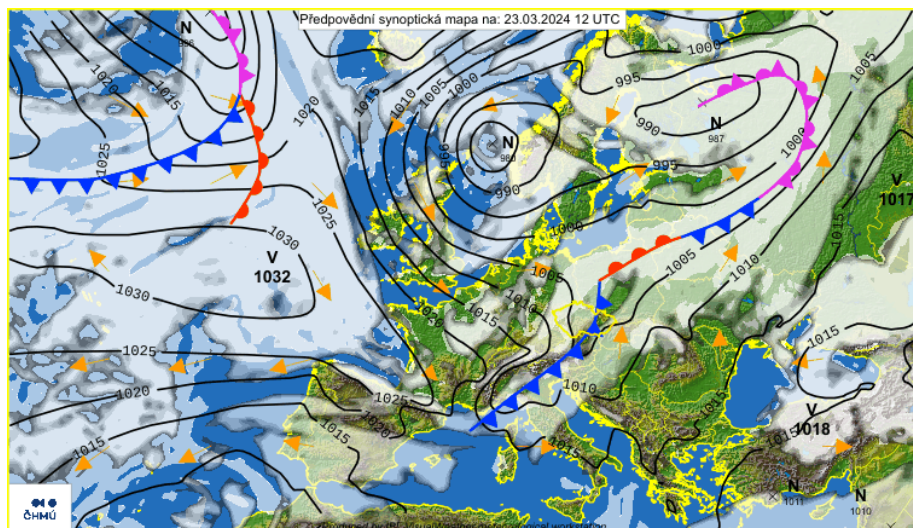
Ve výstupu je ale přítomen i negativní jev, kterým je frontogenetický obrys Grónska. Ten vzniká kvůli výraznému teplotnímu gradientu právě v místech, kde přechází nezamrzlá pevnina do trvalého ledu (obrázek 7).

Tento problém lze částečně vyřešit aplikováním předpokladu, že právě tato nechtěná místa nemívají dostatečné množství vlhkosti ve vzduchových hmotách. Pokud vypočtenou frontogenezi vynásobíme binární maskou, která pokrývá

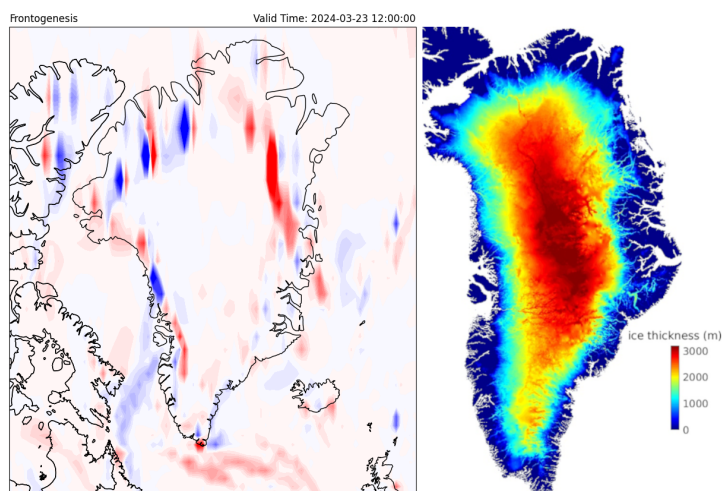
místa s nízkou vlhkostí, dosáhneme malého zpřesnění, které je zobrazeno v porovnání na obrázku 8. Zároveň lze vzít v úvahu, že je toto rozhraní statické a vytvořit pro něj pevnou masku hodnot.



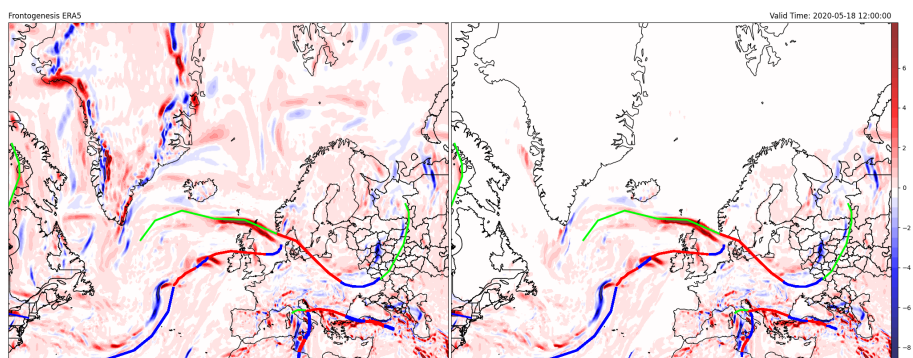
Obrázek 5: Vypočtená frontogeneze z 23. 3. 2024 12:00 UTC



Obrázek 6: Předpovědní synoptická mapa na 23. 3. 2024 12:00 UTC[12]



Obrázek 7: Srovnání frontogeneze nad Grónskem s ledovou pokrývkou

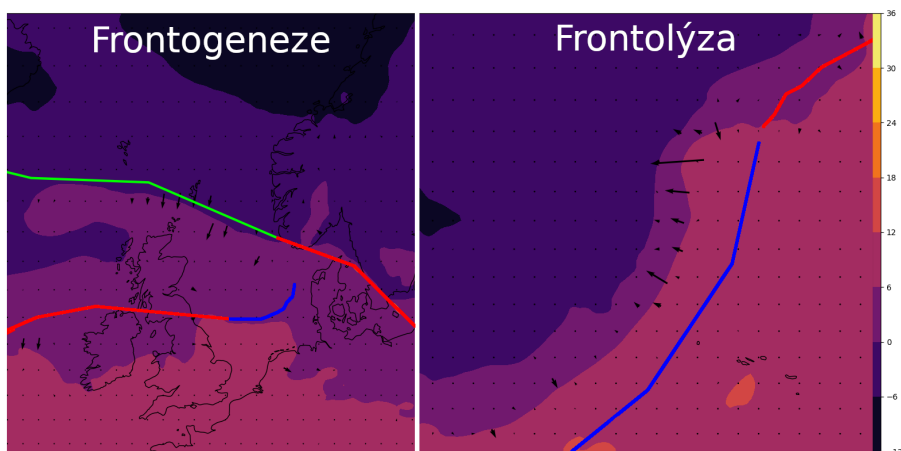


Obrázek 8: Frontogeneze s aplikovanou maskou

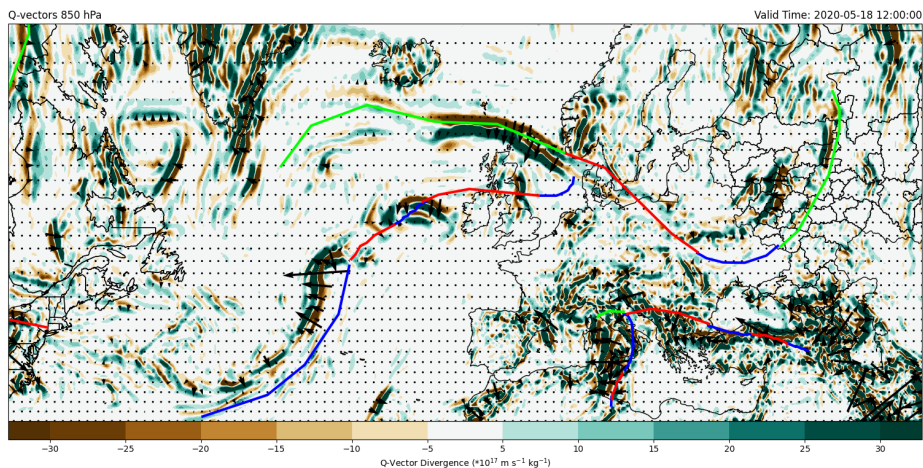
### 2.2.5 Q-vektory

Tyto vektory sice nejsou žádnou fyzikální veličinou, ale pouze matematické vyjádření atmosférických procesů. Značí vektorovou míru změny teplotního gradientu podél izobarické trajektorie. Pro nalezení frontogenetických a frontolytických zón nás bude zajímat směr těchto vektorů. Ten když vede ze studeného vzduchu k teplému, naznačuje frontogenezi, v opačném případě frontolýzu, jak je znázorněno na obrázku 9.

Z těchto vektorů můžeme vypočítat jejich divergenci a vykreslit si ji na obdobnou mapu 10. Zde nám záporné hodnoty ukazují místa, kde se mají vektory tendenci sbíhat a kde tedy dochází k pohybu vzduchových hmot směrem vzhůru na teplé části fronty. Kladné hodnoty naznačují opačný efekt, tedy rozbíhavost a sestupný směr pohybu vzduchových mas na studené straně fronty.[15]



Obrázek 9: Směry Q-vektorů na teplotě v 850 hPa

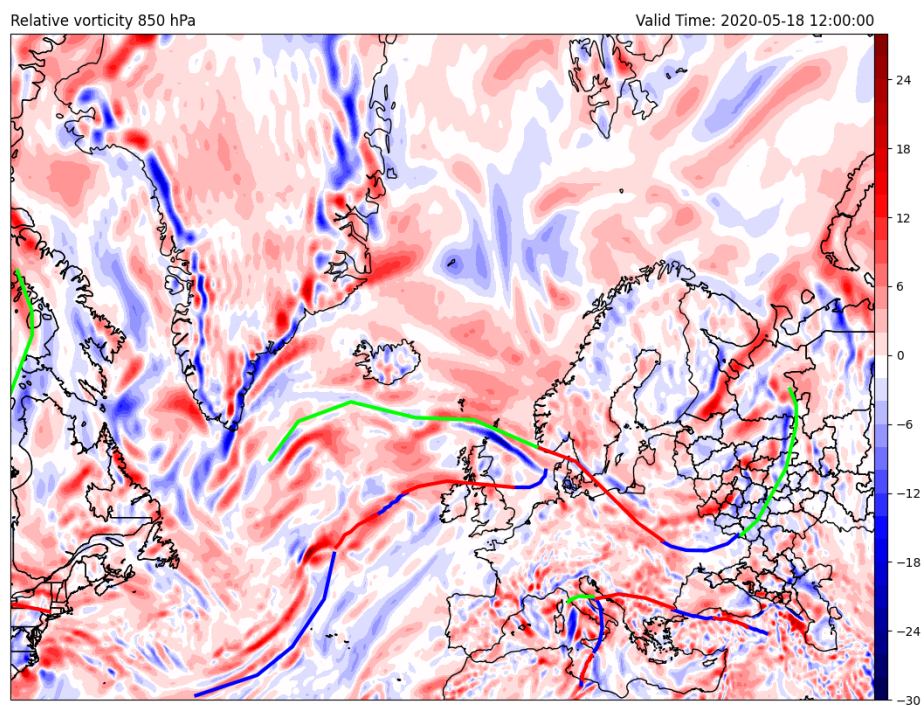


Obrázek 10: Q-vektory zobrazené šipkami a jejich divergence

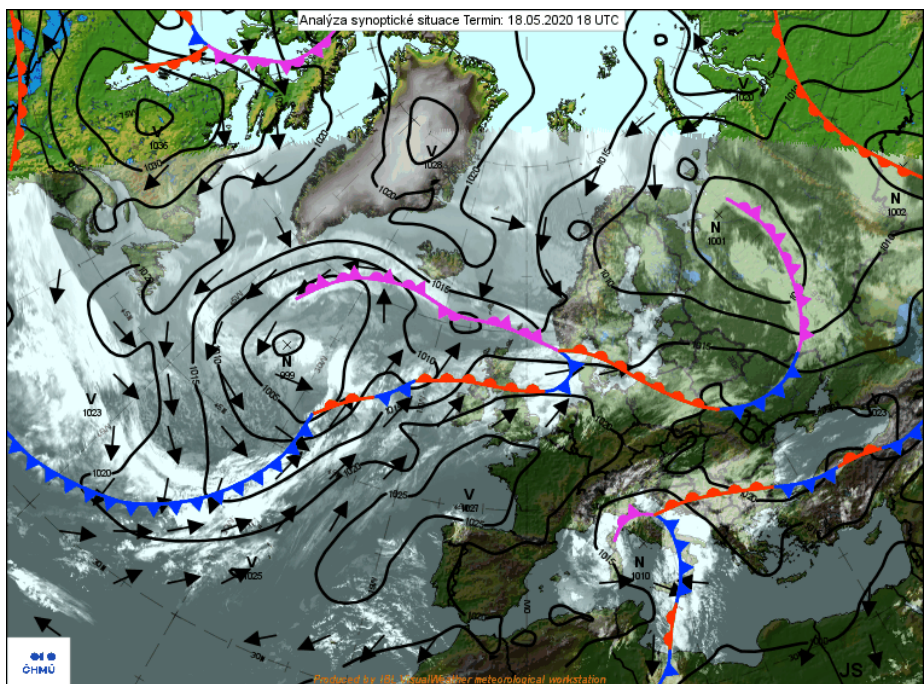
### 2.2.6 Relativní vorticity

Jedná se o vektorovou veličinu, která charakterizuje rotaci vzduchu v synoptickém měřítku. V závislosti na směru rotace vzduchu nabývá buď kladných nebo záporných hodnot. V případě cyklonálního charakteru jsou tyto hodnoty kladné, v případě anticyklonálního jsou záporné. Fronty nám pomáhají detekovat právě vysoké kladné hodnoty, které se vyskytují u cyklón a brázd nízkého tlaku vzduchu. Pro ilustraci je přítomný obrázek 11 s vorticitou a obrázek 12 se synoptickou mapou od ČHMÚ pro porovnání.





Obrázek 11: Relativní vorticity v hladině 850 hPa



Obrázek 12: Anlyza synoptické situace z 18. 5. 2020 [10]

### 2.2.7 Prvky pro subjektivní analýzu

Protože algoritmy pro objektivní analýzu synoptické situace nejsou všemocné a občas produkují nepřesné nebo chybějící linie front, stále se praktikuje manuální zakreslování analýzy synoptické situace. Ty provádí meteorolog pomocí série map z modelových výstupů. Níže jsou vyjmenovány jednotlivé produkty, které vedou k subjektivnímu odhadu pozice front:

- Teplota v hladině 850 hPa
- Rozložení tlakového pole – meteorolog má většinou v kombinaci s jiným produktem na mapě znázorněné tlakové izobary. Frontu můžou indikovat brázdý nízkého tlaku vzduchu.
- Oblačnost, družicové produkty – vlivem kondenzace vynášených vzduchových hmot na frontálním rozhraní vzniká oblačnost, kterou můžeme zpětně použít k identifikaci.
- Srážky – každý typ fronty má své srážkové charakteristiky, proto nám právě srážky pomohou určit výskyt fronty a přiblížit její charakter.
- Vítr v hladině 300 hPa – již popsáno výše.
- Vlhkost mezi hladinami 850 - 500 hPa – právě vlhkost ve vyšších hladinách ukazuje místa, kde je vlhkost vynesena frontálním rozhraním.

## 3 Použité technologie

Při výběru technologií pro tento projekt jsem se zaměřil na tyto věci: stabilitu, univerzálnost a jejich aktivní vývoj. Tyto parametry jsou důležité pro udržení tohoto projektu při životě i po odevzdání SOČ, což jeden z původních plánů. Cílovou platformou jsou primárně operační systémy unixového typu, ale díky použití programovacího jazyka Python je možný nekomplikovaný přenos na jinou platformu.

### 3.1 Python 3

Python je vysokoúrovňový, převážně interpretovaný programovací jazyk, známý svou jednoduchostí, čitelností a elegantností syntaxe díky nucenému psaní indentačních bloků místo složených závorek, jak bývá zvykem u většiny ostatních programovacích jazyků. Disponuje dynamickou typovou kontrolou, což zjednodušuje psaní kódu a flexibilitu, což je praktické zejména ve vědeckých aplikacích, datové analýze a práci se strojovým učením. Díky velké komunitě, která tento jazyk používá, je dostupná široká škála knihoven z různých odvětví. [17]

Nevýhodou oproti jazykům nižší úrovně je zejména výkon, který se ale dá vylepšit použitím knihoven, jako je například Numpy při výpočtech. A to díky tomu, že je tato knihovna částečně implementovaná v jazyce C, tedy na nižší úrovni. Další její výhodou je možnost provádění vektorových výpočtů, které umožňují provádění numerických operací v celém poli najednou.

### 3.2 PyTorch

Pro vývoj v oblasti strojového učení a umělé inteligence je velice vhodné mít nějakou úroveň abstrakce nad reálným fungováním neuronových sítí v pozadí. Jednou z knihoven pro jazyk Python, které jej poskytují, je právě PyTorch. S tímto frameworkem je možné jednoduše vytvářet, trénovat a nasazovat do produkce neuronové sítě s minimálním množstvím kódu. PyTorch rovněž nabízí možnost práce s grafickými procesory (GPU) pro zrychlení výpočtů, což je klíčové pro efektivní zpracování velkých datových sad, jako je například i zde použitá sada obrázků situací. [18]

Na ukázce 1 je znázorněn konvluční blok napsaný s použitím této knihovny. Jedná se o součást generátoru v použité neuronové síti.

```

1 class ConvBlock(torch.nn.Module):
2     def __init__(self, input_size, output_size):
3         super(ConvBlock, self).__init__()
4         self.conv = torch.nn.Conv2d(input_size, output_size,
5             kernel_size=4, stride=2, padding=1)
6         self.lrelu = torch.nn.LeakyReLU(0.2, True)
7         self.bn = torch.nn.BatchNorm2d(output_size)
8
9     def forward(self, x):
10        out = self.conv(self.lrelu(x))
11        return self.bn(out)

```

Ukázka kódu 1: Třída definující konvoluční blok

### 3.3 OpenCV

Jak již název napovídá, jde o široce používanou knihovnu v oblasti počítačového vidění. Je vyvinuta v jazyce C++ a má aplikační rozhraní pro použití v různých programovacích jazycích, včetně Pythonu. Poskytuje celou řadu funkcí pro zjednodušení práce s obrazem či videem. Mezi ně patří detekce objektů, extrakce rysů, analýza a zpracování obrazových dat, segmentace, filtry a mnoho dalšího. [19] V této práci je tato knihovna využita při převodu synoptické mapy do strojového formátu. V ukázce kódu 2 je funkce pro vytvoření binárních masek front podle jejich barev.

```

1 def f_extract(image, color_definition=chmi_colors, tolerance=50):
2     fronts = []
3     # extract fronts by color, create mask
4     for colord in color_definition:
5         fronts.append(
6             cv.inRange(
7                 image, colord - tolerance, colord + tolerance
8             )
9         )
10    return fronts

```

Ukázka kódu 2: Tvorba masek front

### 3.4 GDAL lib

Kvůli rozdílu geografického zobrazení mezi vstupním obrázkem a výstupem je potřeba podniknout kroky zvané georeference<sup>2</sup> a reprojekce<sup>3</sup>. K tomu slouží tato knihovna napsaná v C++, která také poskytuje rozhraní pro Python, ovšem jenom jako bindings generované nástrojem SWIG<sup>4</sup> [20]. Je využívána například v pozadí programu QGIS<sup>5</sup> a existují k ní mocné utility pro příkazovou řádku.

<sup>2</sup>Přiřazení geografického zobrazení obrázku.

<sup>3</sup>Převod mezi geografickými zobrazeními.

<sup>4</sup>Vývojový nástroj pro spojování programů napsaných v C nebo C++ s jazyky vyšší úrovně.

<sup>5</sup>Opensource geografický informační systém.

## 3.5 Skript shellu

Pro spojení více samostatných nástrojů do sebe a automatizaci úloh se používají v unix-like systémech skripty shellu. Jsou to textové soubory obsahující sekvence příkazů pro shell<sup>6</sup>. Umožňují spouštět jiné programy, přeposílat jejich výstupy, používat podmínky, cykly a funkce. Mohou být psány v různých dialektech shellu, jako je například nejznámější Bash. V mém systému je použit k automatizaci konverze obrázků, tvorbě datasetů či rozšíření datasetů. Právě poslední jmenovanou činnost vykonává skript v ukázce **3**. Ten je určený pro shell Bash. Prochází všechny soubory v dané složce a pro každý z nich vykoná:

- k současnému souboru s daty nalezne protějšek se stejným jménem souboru bez koncovky,
- zopakuje for smyčku pro několik různých směrů natočení,
- ve smyčce provede pomocí příkazu convert natočení obou obrázků a uloží je pod změněným názvem souboru.

```
1 #!/bin/bash
2 data_files="datasets/generated_model-v4/data"
3 fronts_files="datasets/generated_model-v4/fronts"
4 files=$data_files/*
5
6 for data_image in $files; do
7     echo $data_image
8     fname=$(basename "${data_image%.}")
9     front_image="$fronts_files/$fname.png"
10
11     for rotation_angle in "90" "180" "270"; do
12         rot_angle=$(shuf -i 0-360 -n 1)
13         data_image_out="$data_files/$fname-$rot_angle.png"
14         fronts_image_out="$fronts_files/$fname-$rot_angle.png"
15         echo "Rotating $fname -> $rot_angle degrees"
16
17         convert "$data_image" \
18             -background white \
19             -rotate "$rot_angle" \
20             "$data_image_out"
21
22         convert "$front_image" \
23             -background white \
24             -rotate "$rot_angle" \
25             "$fronts_image_out"
26     done
27 done
```

**Ukázka kódu 3:** Skript shellu pro augmentaci datasetu

<sup>6</sup>Příkazový interpret umožňující uživateli práci s operačním systémem.

## 3.6 GTK 3

Pro vývoj nástroje pro manuální kontrolu datasetů, vyřazování nevhodných dat a přípravu nových datasetů bylo třeba vytvořit grafické uživatelské rozhraní. Existuje mnoho knihoven a sad nástrojů pro jeho tvorbu, mezi něž patří například QT. Já jsem využil knihovnu GTK, která byla původně vyvinuta jako součást grafického editoru GIMP [21] a dnes se hojně používá pro tvorbu velké části programů pro linuxové distribuce. Obsahuje řadu nástrojů, prostředků a widgetů, ze kterých lze stavět GUI<sup>7</sup> aplikací.

## 3.7 Git

V současnosti je předpokladem pro pokračování a udržitelnost projektu s kódem využití verzovacích systémů. Ty nám umožňují spravovat historii změn a výrazně zjednodušují práci v týmu. Kromě Gitu existují i jiné systémy – mezi ně se řadí například Mercurial a Apache Subversion (SVN). Git jsem zvolil hlavně kvůli snadné dostupnosti hostovacích služeb, jako je GitHub nebo GitLab.

Správa verzí pomocí systému Git umožňuje tvořit bloky změn zvané *commits*, pracovat s jejich historií, vytvářet vývojové větve, posílat požadavky o začlenění změn a mnoho dalšího. Hostovací platformy pak navíc umožňují hlásit problémy či spravovat vydání softwaru.

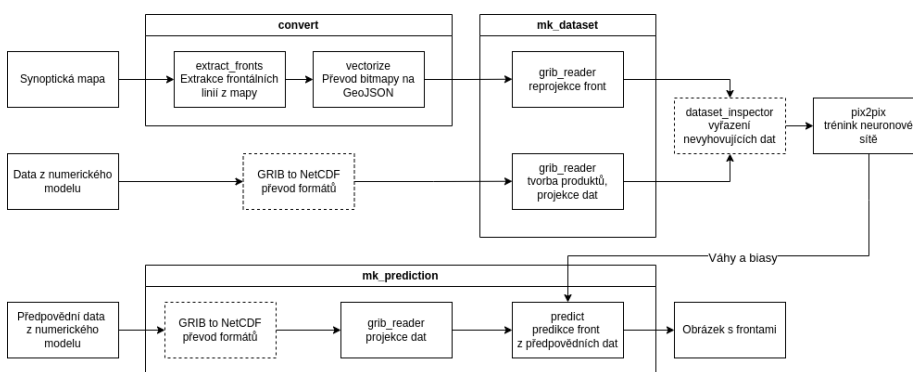
---

<sup>7</sup>Graphical user interface

## 4 Architektura softwaru

Softwarová implementace má za úkol spojit meteorologická data, teorii a z obou těchto složek vytvořit smysluplný výstup ve formě konkrétních linií atmosférických front. Vzhledem k důrazu na modulárnost a možnost jednoduššího využití jednotlivých částí softwaru pro další výzkum kýmkoliv jiným je software rozdělen na dílčí skripty, které vykonávají spolehlivě jen jednu funkci. Jedná se tedy spíše o sadu nástrojů, které mají díky své spolupráci dosáhnout hlavního vytyčeného cíle. Tato vlastnost je inspirována filozofií programů pro systém Unix, která právě na zmiňované vlastnosti apeluje.

Na níže přiloženém obrázku 13 jsou znázorněny jednotlivé části a jejich návaznosti na sebe v rámci celého procesu. Můžete si povšimnout, že naprostou majoritu tvoří nástroje v počáteční části řetězce, které slouží ke zpracování vstupních dat a k přípravě datasetu pro trénink neuronové sítě. Samotná neuronová síť typu GAN tvoří až poslední článek, který je po úspěšném tréninku schopný nezávisle na většině ostatních prvků tvořit predikce na základě předložených dat z numerického modelu. Součástí jsou ale i nástroje, které nejsou v řetězci zaznačeny, protože jejich hlavní účel je testování a diagnostika. Mezi ně patří například skript pro tvorbu map, jež byl použit pro ukázky v sekci *Meteorologická teorie*.



Obrázek 13: Schéma architektury softwaru

### 4.1 Vstup meteorologických dat

Aby bylo možné natrénovat neuronovou síť, je potřeba mít jak vstupní modelová data o počasí, tak očekávaný výstup z nich. Zdrojem meteorologických dat jsou tzv. numerické předpovědní modely, známé hlavně pod zkratkou z anglického názvu *Numerical weather prediction model* – *NWP*, které provozují některé národní meteorologické služby. Jsou to složité algoritmy sestavené z množství fyzikálních rovnic, které jsou velmi výpočetně náročné. Proto, aby bylo možné dostat předpověď před datem její platnosti, je potřeba odpovídající hardware.

Modely se dělí podle regionu, ve kterém se počítají, na globální a regionální. Mezi regionální modely se řadí například Aladin provozovaný Českým hydro-

meteorologickým ústavem. Mezi globální pak například americký GFS<sup>8</sup> nebo ECMWF od Evropského centra pro střednědobou předpověď počasí. Právě poslední jmenovaný je použit jako zdroj pro tuto práci, konkrétně tedy reanalýza za předchozí roky, která byla tímto modelem spočítána. Tato data jsou volně přístupná v rámci projektu *Copernicus Climate Change Service* jako produkt *ERA5* [16].

Ve webové aplikaci *Copernicus Climate Change Service* je možnost si u zvoleného produktu nakonfigurovat, která část dat je pro nás zajímavá, což výrazně zredukuje datový přenos při stahování. Je možné vybrat chtěné proměnné v určitých výškových hladinách a na námi vybraném území. Po odeslání požadavku pro vytvoření tzv. subsetu je naše úloha zařazena do čekací fronty. Vzhledem k nastaveným pravidlům zde mají vyšší prioritu menší požadavky, takže dle vlastní zkušenosti je čekací doba pro data za jeden rok o objemu 16 GB zhruba 4–6 hodin.

## 4.2 Vstup polohy front

Další složkou je zakreslená synoptická situace, které poslouží jako ukazatel reality při tréninku. Jde o znázornění, kde se ve skutečnosti nacházejí linie front. Meteorologické ústavy vytvářejí zpravidla dva typy obrázků.

Jedním z nich je předpovědní synoptická mapa, která je ale založena na předpovědních datech a generována pravděpodobně automaticky. To ji činí nepoužitelnou pro tento projekt. Druhou možností je analýza synoptické situace, která je vytvářena meteorologem a vychází ze v tom momentě aktuálních dat. Díky tomu je nejlepší volbou, se kterou je vhodné při tvorbě tréninkových datasetů pracovat.

Jako zdroj jsem použil synoptické mapy od Českého hydrometeorologického ústavu, které mají dostupně na svém webu [10]. Existuje však vícero poskytovatelů, jako je například *Deutscher wetterdienst* nebo britský *MetOffice*. Všechny tyto zdroje mají ale zásadní nedostatek a sice že nemají databázi situací několik let dozadu, protože na trénink jsou potřeba alespoň nižší tisíce situací.

Tento problém se dá obejít včasným automatickým stahováním, jak bylo učiněno i v tomto případě. V práci je díky tomu použito necelých 3000 analýz od roku 2015. Jelikož jsou tyto mapy kreslené manuálně, respektive pomocí specializovaného softwaru na bázi GIS<sup>9</sup>, tak občas obsahují různé vady, nejčastěji v posunutém pořadí vrstev, nebo změněná velikost textu v hlavičce s informacemi o datu analýzy. To způsobuje problémy v následném procesu extrakce frontálních linií.

## 4.3 Zpracování

Oba datové vstupy je před použitím v trénovacím datasetu třeba předpřipravit. U synoptických dat musí dojít k převodu do strojově čitelného formátu, ze kterého jsou následně vykresleny pod jiným zobrazením do obrázku s reálnou situací. Současně v tomto procesu dochází k určení data a času, kdy synoptická situace nastala. Podle něj se vybere korespondující položka z datasetu NWP modelu. Z těchto dat se provede výpočet meziproductů, které vycházejí z poznatků

<sup>8</sup>GFS = Global forecast system

<sup>9</sup>GIS = Geografický informační systém



kapitoly *Ukazatele výskytu front* a ty jsou následně zasazeny do geografického prostoru a uloženy jako obrázky.

Kvůli variabilní kvalitě dat je v některých případech provést manuální kontrolu vytvořeného datasetu, jak je naznačeno v bloku *dataset\_inspector*. Pro tento účel vznikla stejnojmenná grafická utilita, která manuální proces výrazně zrychluje.

## 4.4 Výstup

Vytvořený dataset může volitelně projít krokem augmentace, v němž dojde k rozšíření o obrázky vzniklé posuvy a otočením původních prvků z datasetu. Následně pomocí obrázku se zakódovanými daty z NWP modelu a s frontálními liniemi probíhá trénink neuronové sítě.

Při něm se průběžně ukládají váhy a biasy v různých stádiích tréninku za účelem výběru nejlepšího výsledku. Tyto uložené váhy a biasy lze pomocí skriptu *predict* načíst a použít je k předpovědi výskytu front podle vstupního obrázku, který na základě předpovědních výstupů NWP modelů připraví již jednou použitý skript *grib\_reader*.

Stejně tak lze teoreticky znovu použít skript *vectorize* pro převod vzniklého obrázku do strojově lépe čitelného formátu. Tato funkce však v současné chvíli není implementována.

## 5 Zpracování obrazových dat

Na úplném počátku cesty od vstupních dat k natrénovanému modelu jsou mapy synoptické situace, které jsou vydávány na webu ČHMÚ jako obrázek určený pro lidské bytosti. Jsou zde zakresleny různé prvky:

- oceán,
- obrysy kontinentů a států,
- oblačnost,
- izobary,<sup>10</sup>
- fronty,
- logo a popisky.

Ovšem nás zajímá pouze jedna položka ze seznamu – fronty a jejich umístění v prostoru. Proto je potřeba použít metody strojového vidění pod knihovnou OpenCV, abychom se zbavili nežádoucích prvků a ve výsledku získali jen souřadnice front rozdělené podle jejich typů. Tento proces se skládá z několika kroků, které níže popíši.

### 5.1 Extrakce front

V prvním kroku zpracování obrázků, jako je například obrázek 1, se na obrázek aplikuje maska, která zakryje rušivé prvky. Mezi ně se řadí logo a vodoznak, který říká v jakém systému byl obrázek vytvořen. Následně proběhne výběr rozmezí pixelů hlavičky s datem a časem platnosti analýzy. Z této matice pixelů se pomocí OCR<sup>11</sup> Tesseract rozpozná text. Tento řetězec je dále zpracován, abychom z něj získali časové razítko, které bude později uloženo do EXIF hlavičky obrázku s čistými liniemi.

Následuje samotná extrakce linií. Provádí se tak, že jsou podle předem definovaných barev vybrány všechny pixely v určité toleranci a takto vzniklé masky jsou uloženy do polí podle charakteru fronty – teplá (červená), studená (modrá), okluzní (fialová). Následně je každá z front rozdělena na její jednotlivé úseky. Každý z nich projde procesem *thinningu*<sup>12</sup>, při kterém je sérií dilatací a erozí přeměněn na část linky.

```
1 def extract_front_line(mask_in, nr_iterations=10):
2     skeleton_new = np.zeros_like(mask_in)
3     num_labels, labels = cv.connectedComponents(mask_in)
4
5     for label_id in range(num_labels):
6         mask = (labels == label_id + 1).astype("uint8") * 255
7
8         mask = cv.dilate(mask, kernel, iterations=2)
```

<sup>10</sup>Čáry spojující místa se stejným tlakem.

<sup>11</sup>Optical character recognition.

<sup>12</sup>Proces ztenčování široké skupiny pixelů na šířku 1 px.

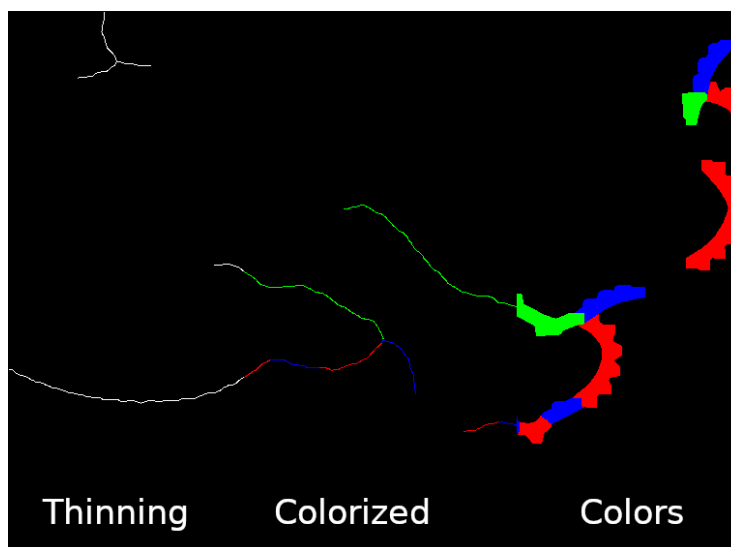
```

9     spikes = cv.erode(mask, kernell1, iterations=1)
10
11     skeleton = cv.ximgproc.thinning(mask, 1)
12     skeleton_erodet = np.copy(skeleton)
13
14     for i in range(nr_iterations):
15         skeleton_erodet = cv.dilate(skeleton_erodet,
16                                     kernell1,
17                                     iterations=1)
18         skeleton_erodet = cv.erode(skeleton_erodet,
19                                     kernell2,
20                                     iterations=1)
21
22     skeleton_new += cv.ximgproc.thinning(skeleton_erodet, 1)
23     return skeleton_new

```

**Ukázka kódu 4:** Zjednodušená funkce pro extrakci linií z masky fronty

V konečné fázi je ze všech úseků sestavena nová maska, která se obarví podle dilatovaných barev původního obrázku operací *bitwise and*. Všechny kroky jsou zobrazeny na obrázku 14.



**Obrázek 14:** Kroky při extrakci front

## 5.2 Georeference

Nyní máme každou frontu jako jeden pixel širokou linku na stejných pozicích v obrázku jako originál. Aby bylo možné obsah obrázku promítnout na povrch země, musíme znát tzv. georeferenční body. Kdybychom znali, kde se nachází každý z rohů obrázků a k tomu parametry původního zobrazení, máme vyhráno. Bohužel však tyto informace nemáme, takže musíme zvolit metodu manuální georeference. K tomuto účelu se používají geografické informační systémy, což jsou programy pro práci s mapovými vrstvami a daty. Při georeferenci obrázků jsem použil QGIS.

Zde existuje nástroj zvaný rastrový georeferencér, do kterého si člověk nainportuje potřebný obrázek a postupně vybírá body z obrázku a umísťuje je na mapu. Typicky se jako tyto body používají hranice států a zřetelné prvky v obou zobrazeních. Následně po spuštění algoritmu georeference<sup>13</sup> dojde k umístění rastru na mapu ve zvoleném zobrazení. Výhodou je možnost exportu skriptu pro GDAL se všemi body a parametry transformace **5**, který se následně používá v převodu do GeoJSONU.

```
1 gdal_translate -of GTiff \  
2   -gcp 379.603 508.592 -1.43576e+06 832021 \  
3   -gcp 427.764 260.015 -587902 3.47698e+06 \  
4   -gcp 344.684 193.858 -1.50661e+06 4.36171e+06 \  
5   input output  
6 gdalwarp -r bilinear -tps -co COMPRESS=NONE  
7   -t_srs "+proj=eqdc +lat_0=30 \  
8   +lon_0=10 +lat_1=43 +lat_2=62 \  
9   +x_0=0 +y_0=0 +ellps=intl +units=m +no_defs" \  
10  input output
```

Ukázka kódu 5: Skript pro georeferenci pomocí GDAL

## 5.3 Strojově čitelný formát

Jako strojový formát jsem zvolil GeoJSON kvůli jeho široké použitelnosti a kompatibilitě s mnoha nástroji a GISy. Jde o standardní JSON, která má přizpůsobenou strukturu pro zaznamenání geografických informací. Ve svém programu zaznamenávám fronty jako lomené čáry, k čemuž se používá kombinace struktury *FeatureCollection*, uvnitř které je *features* s geometrií *LineString*, jak je znázorněno na ukázce **6**.

```
1 {  
2   "type": "FeatureCollection",  
3   "properties": {  
4     "datetime": 1483268400,  
5     "type": "chmi_analysis",  
6     "projection": "original",  
7     "note": "test_2024_02_24_nooverlay"  
8   },  
9   "features": [  
10
```

<sup>13</sup>Ten v QGISU probíhá pomocí GDAL.

```

10     {
11         "type": "Feature",
12         "geometry": {
13             "type": "LineString",
14             "coordinates": [
15                 [
16                     -45.863723,
17                     52.728575
18                 ],
19                 ...
20                 [
21                     -50.140054,
22                     61.358529
23                 ]
24             ]
25         },
26         "properties": {
27             "front_type": "cold",
28             "front_id": 0
29         }
30     },
31     ...
32 ]
33 }

```

**Ukázka kódu 6:** Ukázka GeoJSONu

Před konverzí do tohoto formátu musí nejdříve nastat rozložení jednotlivých linií na vícero úseček a převod souřadnic. První proces funguje tak, že se k obrázku nejdříve vytvoří georeferencovaná kopie ve formátu GeoTIFF<sup>14</sup>, která se uloží. Následně je obrázek rozporcován na samostatné úseky front dle jejich typu a odfiltrují se nežádoucí shluky pixelů, které jsou menší než prahová hodnota. S každým úsekem fronty se pracuje jako s binární maskou a určují se souřadnice všech jeho pixelů. Souřadnice jsou následně seřazeny a jejich počet mezi počátkem a koncem linie je zredukován.

Poté jsou ke všem bodům zjištěny jejich geografické souřadnice právě za pomoci transformací z GeoTIFFu. Z těchto souřadnic jsou vytvořeny dvojice s indexy  $(0, 1)$ ,  $(1, 2)$ ... a zapsány do GeoJSONu.

<sup>14</sup>Formát georeferencovaného rastrového obrázku.

## 6 Zpracování dat z NWP modelů

Dalším stupněm softwaru je část, která se stará o generování prvků datasetu s meteorologickými daty. Ta pocházejí z numerických modelů a jsou distribuována ve dvou hlavních formátech:

- GRIB – Jde o formát definovaný WMO<sup>15</sup>. Soubor GRIB může obsahovat několik tzv. *messages*, které jsou na sobě nezávislé a mohou být jakkoliv uspořádány. Každá z nich potom obsahuje několik sekcí s metadaty informujícími o tom, jaká data se uvnitř nacházejí a jak mají být promítnuta na zemský povrch, a samotná mřížková data. [22]
- NetCDF – Na rozdíl od předchozího formátu je NetCDF uspořádáno hierarchicky do struktury vícerozměrových polí. Informace jsou rozčleněny do proměnných (teplota, vlhkost, ...) a dimenzí, což definice zeměpisné šířky, výšky, hladiny a času. Vše je uspořádáno tak, aby byla data sebepopisující. Vzhledem k tomu, že práce s velmi velkými datasety v GRIBu je s použitím dostupných knihoven pro jazyk Python pomalá, byl pro uchovávání meteorologických dat zvolen právě formát NetCDF.

### 6.1 Práce s daty

K načtení datasetu se používá knihovna *xarray*, která si při otevření načte rozměry a proměnné. Samotná data jsou přenášena do paměti až poté, co je zavoláno čtení samotného datového pole nebo jeho části. Ke zjednodušení fyzikálních výpočtů se používá knihovna *MetPy*, která je přímo určena pro aplikaci v oboru meteorologie. Při manipulacích s daty spolupracuje s knihovnou *pint*, která se stará o jednotky a převod mezi nimi, jak je znázorněno v ukázce 7.

```
1 ds = xr.open_dataset(file).metpy.parse_cf()
2 # nastaveni hladiny s jednotkou
3 frontogenesis_level = 850 * units.hPa
4 # nacteni teplot ve vsech hladinach v datum nejblize target_date
5 t = ds['t'].metpy.sel(method='nearest', time=target_date)
6 # vyber tlakove hladiny
7 t_850 = t.sel(level=frontogenesis_level,
8              method='nearest').metpy.unit_array.squeeze()
9 # vypocet potencialni teploty z teploty
10 pt_850 = mpcalc.potential_temperature(frontogenesis_level, t_850)
11 # prevod jednotek
12 temperature_celsius_850 = pt_850.to('degC')
```

Ukázka kódu 7: Práce s datasetem pomocí *xarray* a *MetPy*

<sup>15</sup>Světová meteorologická organizace.

## 6.2 Složení produktu

Nyní potřebujeme vytvořit pomyslný balíček dat do tréninkového datasetu, který vhodně popisuje stav atmosféry tak, aby z něj byla neuronová síť schopna odvodit pozici a typy front. Aby mohla nastat generalizace, musíme najít takové datové vstupy, které si jsou co nejvíce podobné v rozličných místech výskytu front.

Podle kapitoly *Meteorologická teorie* máme na výběr poměrně širokou škálu možností, ovšem s tím, že se ale v různých funkcích pro výpočet frontogenetických indikátorů opakují stejné vstupní veličiny. Pro co nejlepší výsledky je dobré zvolit co nejdiversifikovanější kombinaci.

V úvahu připadá určitě teplota v hladině 850 hPa, protože reprezentuje charakter vzduchových hmot, jsme z něj schopni hrubě odhadnout místo výskytu a charakter fronty. Velmi podobně tyto parametry indikuje i vlhkost ve stejné výšce, proto ji nemá smysl s teplotou kombinovat, protože by nepřinesla žádné zpřesnění. Další parametr na zvážení je vítr v hladině 300 hPa, ten se spolehlivě vyskytuje nad rozhraními vzduchových hmot. Dále je zde oblačnost a srážky, ale ty nesou zásadní nevýhodu v tom, že se vyskytují v různé míře i mimo frontální rozhraní. Z vypočtených parametrů jsou tedy nejlépe použitelné  $Q$ -vektory spolu s jejich divergencí a frontogenetická funkce.

Z této úvahy můžeme vyvodit závěr, že zřejmě nejlepší kombinace se bude skládat z následujících prvků:

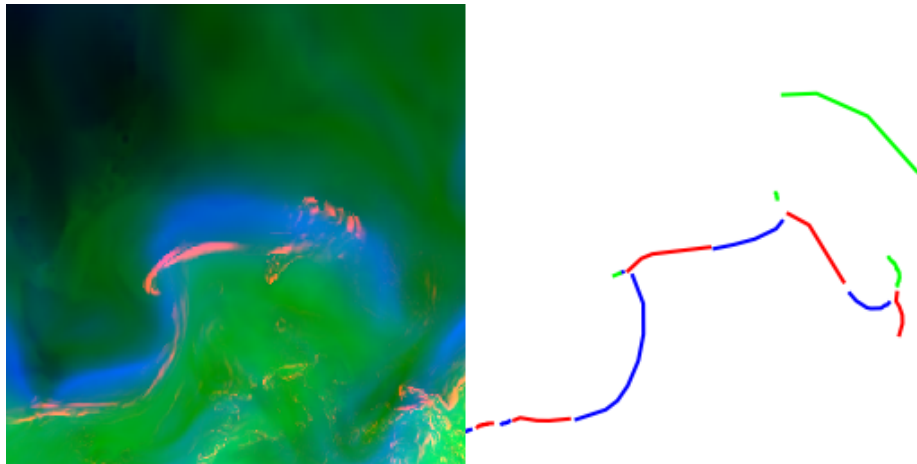
- teplota v hladině 850 hPa,
- vítr v hladině 300 hPa,
- divergence  $Q$ -vektorů nebo frontogenetická funkce ošetřená maskou vlhkostí.

## 6.3 Konverze na obrázek

Pro jednoduchost a snadnou laditelnost v raných stádiích projektu jsem zvolil pro uchování produktů pro trénink rastrové obrázky. Rozhodně nejde o nejlepší řešení kvůli velmi vysokým ztrátám při kódování dat a kvůli limitaci počtu vrstev, které lze využít. Zde jsme limitováni počtem kanálů v obrázku – R, G, B, případně Alpha. Jelikož obrázkové formáty mají omezenou barevnou hloubku, je potřeba provést normalizaci hodnot do tohoto intervalu. Jelikož u ukazatelů výskytu front nás zajímají hlavně kontrasty mezi hodnotami veličin, můžeme zvolit lineární rozvrstvení hodnot mezi minimem a maximem.

Jelikož vypočtené hodnoty jsou uloženy v dvojrozměrném poli, samy o sobě se nedají promítnout na povrch země, protože ve skutečnosti by byly buňky s hodnotami různě velké v závislosti na geografické projekci. K vytvoření obrázku bylo využito rozšíření *basemap* ke knihovně *matplotlib*, které nám umožní jednoduchou cestu k projekci jak meteorologických dat, tak front.

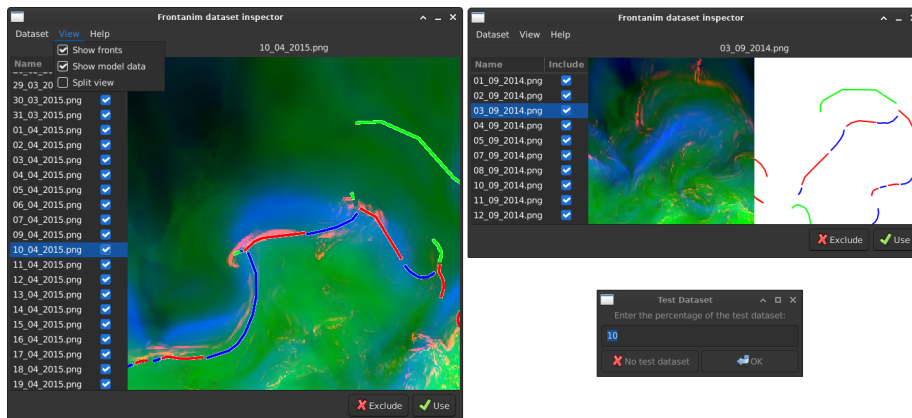
Při tvorbě datasetu pro trénink jsou tedy vytvořeny dva obrázky – první s meteorologickými daty zakódovanými do barevných kanálů a druhý s vykreslenými liniemi front. Ukázány jsou na obrázku 15.



Obrázek 15: Vygenerovaný pár obrázků pro trénink

## 6.4 Výběr validních dat

Ne všechny vygenerované páry obrázků na sebe správně pasují buď kvůli časové odchylce, nebo kvůli špatnému uspořádání vrstev<sup>16</sup> ve vstupním obrázku analýzy z webu. Z tohoto důvodu vznikl nástroj *data\_inspector*. Má za úkol co nejvíce pomoci při manuálním třídění díky minimalizaci kliknutí uživatelem. Nabízí jednoduchá uživatelské rozhraní napsané s použitím GTK, jehož hlavním prvkem je boční lišta se seznamem souborů spolu s náhledovým oknem. To uživateli umožní po výběru složky s datasetem pomocí pouhých dvou tlačítek vyřazovat nevhodné kandidáty a zároveň posouvat dále. Umožňuje také export vyřazeného datasetu včetně souboru se soupisem vyřazených snímků a disponuje funkcí pro automatické oddělení trénovacích dat od testovacích. Náhled GUI je na obrázku 16.



Obrázek 16: GUI aplikace Dataset inspector

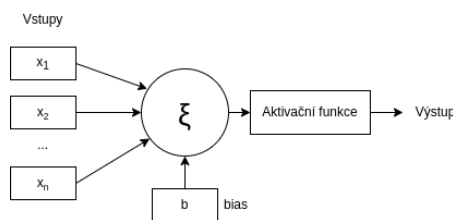
<sup>16</sup>Tato situace nastává nejčastěji když meteorolog omylem umístí izobary přes fronty.



## 7 Neuronová síť

Jak již bylo dříve zmíněno, k plnění hlavní viditelné funkce byl místo běžnějšího hard computingu použit soft computing, konkrétně ve formě neuronové sítě.

Neuronová síť by se dala definovat jako výpočetní model inspirovaný biologickým nervovým systémem, skládající se ze základních propojených jednotek – neuronů. Samotný matematický neuron znázorněný na obrázku 17 má několik vstupů, které plní funkci biologických dendritů.



Obrázek 17: Zjednodušený model neuronu

Každý vstup  $x$  má určenou svou váhu  $w$ , na základě které se koriguje jeho propustnost. Uvnitř neuronu se z těchto vstupů vypočítá vážená suma, ke které se přičte konstanta bias  $b$ .

$$\xi = b + \sum_{i=1}^n w_i \cdot x_i$$

Tento výsledek ( $\xi$ ) se nazývá *potenciál neuronu* a dále se používá jako argument pro *aktivační funkci*, která z potenciálu vypočítá výstupní hodnotu neuronu. Právě zde se projevuje bias jako posun průběhu aktivační funkce po ose  $x$ . Potenciál neuronu a vypočítaná výstupní hodnota mají mezi sebou nelineární vztah. [23][24] Jako aktivační funkce se v mnohých využitích, včetně toho mého, používá *sigmoid* nebo *ReLU*.

Abychom byly schopni modelovat komplexnější vztahy mezi vstupy a výstupy, propojujeme jednotlivé neurony mezi sebou do sítě. Ta má v našem případě počet vstupů podle počtu pixelů ve vstupním obrázku, skryté neuronové vrstvy a opět několik výstupů, kterých je v tomto případě stejně jako vstupů.

Trénink neuronové sítě spočívá v postupné úpravě vah a biasů neuronů pomocí metody backpropagation<sup>17</sup>, aby se dosáhlo co nejnižší hodnoty ztrátové funkce. Ta pracuje s generovaným výstupem a očekávaným výstupem a na základě algoritmu pro zjištění jejich podobnosti vrací určitou hodnotu.

### 7.1 Pix2pix

Při predikci výskytu meteorologických front jsem využil koncept neuronové sítě nazvané *pix2pix*. Tato síť je založena na principu generativní adversariální sítě (GAN) a její hlavní funkcí je transformace jednoho obrázku na druhý. Autoři

<sup>17</sup>Metoda pro trénink neuronových sítí aktualizující váhy od výstupních vrstev směrem ke vstupním.

tuto síť prezentují jako nástroj pro mnoho aplikací, mezi nimiž se nachází i generování mapových dlaždic ze satelitních snímků. [25]

Princip fungování spočívá v současném trénování dvou hlavních částí, což jsou v podstatě dvě nezávislé neuronové sítě:

- Generátor: snaží se vytvořit výstupy, které jsou co nejvíce podobné ground-truth obrázkům.
- Diskriminátor: má za úkol odhalit, jak moc je pravděpodobné, že je jeho vstup reálný obrázek.

Během tréninku se oba tyto modely učí prostřednictvím vzájemného souboje. Generátor se snaží produkovat co nejlepší možné výstupy, zatímco diskriminátor se snaží identifikovat, jestli jsou tyto výstupy produktem generátoru. Tento proces tréninku probíhá iterativně, až se dosáhne určité úrovně kvality výstupů.

### 7.1.1 Generátor

Generátor v *pix2pix* je konvoluční<sup>18</sup> neuronová síť založená na architektuře U-Net.[26] Skládá se z enkodéru, dekodéru a přechodového mostu mezi nimi, jak je znázorněno na obrázku 18.

Funkcí enkodéru je extrakce vlastností z obrázku, čehož je dosaženo pomocí několika konvolučních bloků o velikosti jádra 2x2 pixely. Zde dojde k zmenšení horizontálních a vertikálních rozměrů na polovinu, přičemž se zdvojnásobí počet kanálů. Po každé konvoluci následuje aktivační funkce Leaky ReLU, která zanáší do sítě nelinearitu a pomáhá se zlepšením generalizace dat. Každý výstup z aktivační funkce je přenesen pomocí tzv. *skip connections* na protější vrstvu v dekodéru.

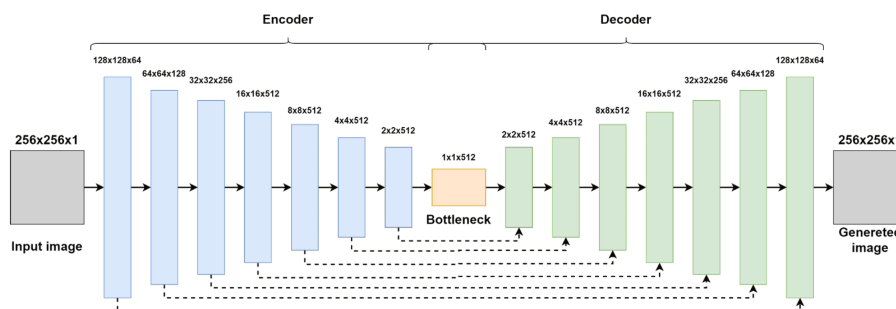
V místě dosažení jednotkových horizontálních a vertikálních rozměrů na enkodér navazuje dekodér, který má následně z úkol obnovit původní rozměry. To se provádí jako série po sobě jdoucích transponovaných konvolucí<sup>19</sup> opět o jádře 2x2. V každém stupni je ke vzniklým tensorům přidána mapa příznaků ze *skip connections* a následně je aplikována aktivační funkce ReLU. [28]

### 7.1.2 Diskriminátor

Diskriminátor v *pix2pix* také pracuje s konvolucí a je založena na architektuře PatchGAN. Výstup generátoru spolu se vstupem meteorologických dat jsou nejprve rozděleny na několik malých oblastí, které se nazývají *paths* a na nich je poté, podobně jako tomu je u enkodérové části generátoru, provedena série konvolucí následovaných aktivační funkcí ReLU. Po zpracování všech *paths* dostaneme skóre reálnosti, které je statisticky zpracována a použita pro přizpůsobení generátoru, případně v kombinaci se vstupními reálnými daty také sama sebe. [28]

<sup>18</sup>Dvourozměrná konvoluce je proces, při kterém se pomocí konvolučního jádra (masky) počítá z původního obrázku nový obrázek s redukováným počtem pixelů, na základě sumy násobků původních hodnot a vah v jádře.

<sup>19</sup>Konvoluce umožňující up-sampling dat.



Obrázek 18: Architektura generátoru [29]

## 7.2 Proces tréninku

Trénink neuronové sítě probíhá tak, že jsou několikrát ve smyčce proběhnuty všechny prvky datasetu. Tyto průběhy se nazývají epochy. V každé z epoch se provádí u každého z obrázků tyto kroky:

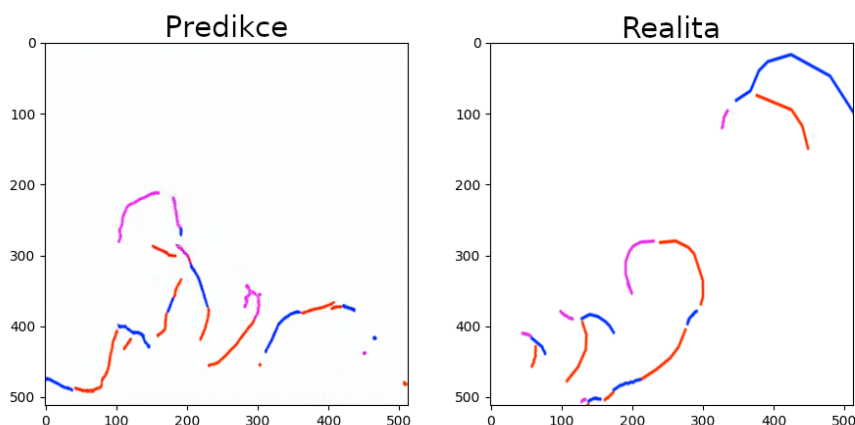
1. Trénink diskriminátoru reálnými daty
  - Nuluje se gradient diskriminátoru.
  - Získá se úsudek diskriminátoru pro reálná data – meteorologické ukazatele a skutečné linie front.
  - Spočítá se ztráta pomocí funkce binární křížové entropie oproti poli samých jedniček, protože se zde předpokládá, že jsou data reálná.
2. Trénink diskriminátoru falešnými daty
  - Pomocí generátoru se vygeneruje falešný obrázek.
  - Pro tento obrázek se získá opět úsudek diskriminátoru.
  - Pomocí binární křížové entropie se spočítá ztráta v porovnání s polem samých nul. Ty nám značí, že tato data neznačí skutečnost.
3. Aktualizace diskriminátoru
  - Spočítá se celková ztráta diskriminátoru jako součet ztráty s reálnými a falešnými daty.
  - Provede se zpětná propagace ztráty.
  - Pomocí optimizéru<sup>20</sup> se aktualizují váhy diskriminátoru.
4. Trénink generátoru
  - Nuluje se gradient generátoru.
  - Pro generovaný obrázek v předchozím kroku se získá úsudek aktualizovaného diskriminátoru.
  - Pomocí binární křížové entropie a pole jedniček se spočítá ztráta značící, jak moc je obrázek reálný.

<sup>20</sup>Jako optimizér je v případě generátoru i diskriminátoru použit algoritmus Adam.

- Spočítá se ztráta mezi vygenerovanými liniemi front a reálnými liniemi. Problematika konkrétních ztrátových funkcí bude vysvětlena v následující kapitole.
- Vypočítá se celková ztráta jako součet předchozích dvou ztrát.
- Dojde ke zpětné propagaci ztrát a optimizér aktualizuje váhy.

### 7.3 Ztrátová funkce

Během tréninku neuronové sítě podle vzoru pix2pix jsem narazil na problém v tom, že model jako výstupy generoval linie, ale tak, že jich bylo mnoho a byly velmi nevhodně distribuovány – viz obrázek 19. Navíc se stávalo, že po cca stovce epoch došlo k výraznému zrnění, které následně vedlo k výraznému růstu diskriminátorové složky ztráty generátoru. Symptomy tedy poukazují na možnou nedostatečnost původní funkce pro ztráty mezi reálnými a generovanými liniemi front.



Obrázek 19: Problematická predikce

Původní použitá ztráta byla  $L1$  loss, což je vlastně průměr absolutních odchylek hodnot pixelů v každém kanálu mezi reálným a generovaným obrázkem. Toto může být praktické u velkoplošných záležitostí, ale v případě hledání linií je výhodnější sáhnout po lepší variantě, která bude pracovat se vzdálenostmi místo s konkrétními pixelovými hodnotami.

Zvolil jsem použití ztrátové funkce za použití *hausdorffovy vzdálenosti*, která je definovaná jako vzdálenost mezi bodem z jedné sady bodů a nejbližším bodem z druhé sady bodů. V programu je tato metoda implementovaná pro každý kanál obrázku zvlášť a v úvahu bere všechny body, které mají hodnotu větší než 0,5 po normalizaci, což jsou právě linie front. Pracuje se tedy s binární maskou front jednoho druhu. Pro každou masku je vypočítáno pole vzdáleností od logických jedniček a odchylka reality a predikce jako absolutní hodnota jejich rozdílů. V poslední řadě je pomocí polí vzdáleností a odchylek vypočtena samotná ztráta. Proces je znázorněn úryvkem kódu 8. Na obrázku 20 je potom znázorněno zlepšení kvality generovaných dat po aplikování této změny.

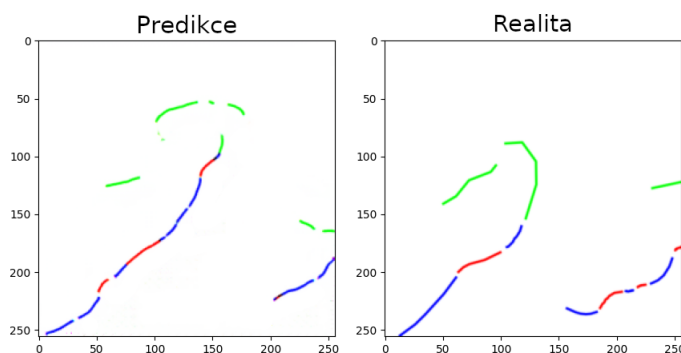
Je nutno ještě podotknout, že podle efektu na data jsem nakonec zvolil kombinaci původní  $L1$  loss a nové  $hausdorff$  loss v poměru 5 : 8. Využití samotné  $hausdorff$  loss mělo totiž za efekt snížení přesnosti predikce charakteru fronty.

```

1 df_pred = distance_field(pred)
2 df_real = distance_field(real)
3 pred_error = (pred - real) ** 2
4 distance = df_pred ** 2 + df_real ** 2
5 dt_field = pred_error * distance
6 loss = dt_field.mean()

```

Ukázka kódu 8: Programové znázornění výpočtu Hausdorffovy ztráty

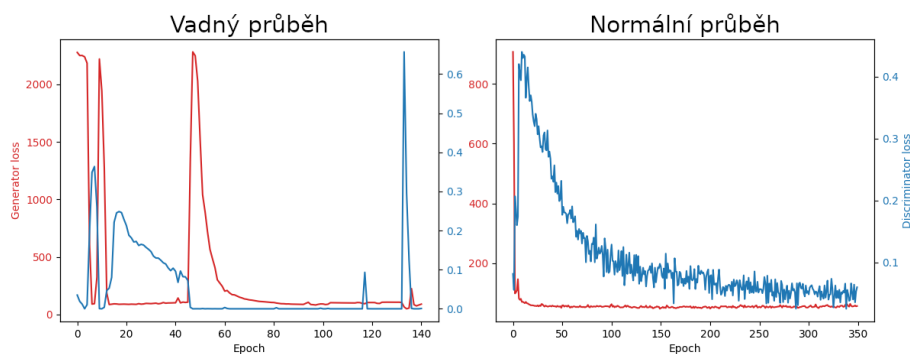


Obrázek 20: Srovnání po úpravě ztrátové funkce

## 7.4 Destrukce kvality

Dalším jevem, který v procesu tréninku nastával, bylo prudké zhoršení kvality predikcí po několika desítkách epoch. Investigoval jsem tedy průběhy ztrát, které jsou znázorněny na levé části obrázku 21. Tady můžeme vidět velmi nekonzistentní průběh učení, kdy na jeho počátku vidíme nízké hodnoty ztráty diskriminátoru, které během plynoucích epoch tvoří nárazovité vrcholky. Zároveň ztráta generátoru v případě nízké ztráty diskriminátoru narůstá, což je očekávaný jev, protože do ztráty generátoru se připočítává úsudek diskriminátoru. Značí nám to tedy, že již v počátečních fázích se generátor natrénoval příliš rychle.

Řešením tohoto problému je přizpůsobení hyperparametrů pro optimizér učení. Konkrétně jde o  $learning\_rate$ , který značí, jak moc se budou v každém kroku učení měnit parametry modelu. S tímto se pojí i možnost využití  $learning\_rate\ schedulerů$  pro možnost tzv. zahřátí – postupné zvedání  $learning\ rate$  během několika počátečních epoch.



Obrázek 21: Srovnání průběhů ztrát při tréninku

## 7.5 Chybějící fronty

Další palčivou záležitostí je nízká schopnost predikovat fronty, které se nacházejí v nejsevernější čtvrtině obrázků. Děje se tak z důvodu nedostatku podobných situací v tréninkovém datasetu. Tento problém částečně řeší augmentace dat, tedy trénink na datasetu rozšířeném o další situace, které však vznikly modifikací již existujících. Nástroje pro tvorbu datasetů tuto funkci mají integrovanou, ale bohužel jediný trénink modelu na taktovém datasetu, který se stihl vykonat před odevzdáním této práce, neprokazuje výrazně lepší schopnosti.

Z meteorologického hlediska se s augmentací může pojít problém s tím, že některé fyzikální parametry mohou být výrazněji ovlivněny zemským povrchem a tím pádem by nebylo vhodné trénovat model na datech, která vznikla současným způsobem augmentace – rotací. Navíc by platilo, že již naučený model bude méně přesný při aplikaci na jiné území.

## 8 Úspěšnost

Na validaci je použita oddělená část datasetu, která nebyla použita k trénování modelu. Pro každou ze situací je vytvořen obrázek predikce a následně je tento obrázek spolu s pozemní realitou použit jako vstup do hodnotícího algoritmu.

### 8.1 Metodika

Algoritmus kombinuje *hausdorffovu vzdálenost* a *teorii detekce signálů*. Pro každý segment fronty v jednom obrázku jsou definovány dvě verifikační kategorie – detekováno a nedetekováno. Ty jsou vyhodnoceny pomocí hausdorffovy vzdálenosti mezi jedním segmentem z prvního obrázku a všemi liniemi front z druhého obrázku. Tento proces je následně použit reverzně, čímž získáme informaci o tom, jestli se jedná o existující, nebo falešnou frontu. Na základě teorie detekce signálů budou tyto výstupy vytříděny do tří kategorií, jak je znázorněno v tabulce 1. Pro rozřazení je využita konstanta  $t$ , která značí maximální možnou hausdorffovu vzdálenost  $d_h$ , při které jsou úseky front ještě brány jako překrývající se.

	Fronta detekována	Fronta nedetekována
Fronta existuje	Zásah (HIT)	Chyba (MISS)
Fronta neexistuje	Falešný poplach (FA)	Správné vyloučení <sup>21</sup>

Tabulka 1: Přehled verifikačních kategorií

Jednotlivé položky tedy znamenají následující:

- zásah – pro  $d_h$  mezi segmentem fronty reálné situace a všemi frontami toho druhu v predikované situaci platí:  $d_h \leq t$ ,
- chyba – pro  $d_h$  mezi segmentem fronty reálné situace a všemi frontami toho druhu v predikované situaci platí:  $d_h > t$ ,
- falešný poplach – pro  $d_h$  mezi segmentem predikované fronty a všemi frontami toho druhu v reálné situaci platí:  $d_h > t$ .

Z těchto položek jsou dále vypočítány verifikační kritéria pomocí vztahů v tabulce 2. [30]

<sup>21</sup>Tato kategorie nemá při hodnocení detekce front smysl, protože by byla aplikovatelná pouze na prázdná místa, pro která však není objektivní měřítko jež by se dalo využít i na zobrazené linie.

Kritérium	Kód	Vzorec
Přesnost	A	$A = \frac{C_{HIT}}{C_{HIT} + C_{MISS} + C_{FA}}$
Zkreslení	BIAS	$BIAS = \frac{C_{HIT} + C_{FA}}{C_{HIT} + C_{MISS}}$
Pravděpodobnost detekce	POD	$POD = \frac{C_{HIT}}{C_{HIT} + C_{MISS}}$
Poměr falešných poplachů	FAR	$FAR = \frac{C_{FA}}{C_{HIT} + C_{FA}}$

**Tabulka 2:** Verifikační kritéria predikce

Zde proměnná  $C_k$  vyjadřuje počet úseků front klasifikovaných do kategorie  $k$ .

## 8.2 Výsledky

Zvolená metodika byla aplikována na testovací dataset obsahující 282 situací. Verifikační kritéria byla spočítána jak pro každý typ fronty zvlášť, tak pro kombinaci všech jejich druhů. Jako model generátoru byl vybrán ten, který během průběžných testů při tréninku prokazoval nejlepší výsledky. Pracuje se s  $t = 3$ .

Prvek	A	BIAS	POD	FAR
<b>Studená fronta</b>	0,46	0,88	0,60	0,33
<b>Teplá fronta</b>	0,48	0,92	0,62	0,33
<b>Okluzní fronta</b>	0,57	0,89	0,69	0,22
<b>Všechny typy</b>	0,52	0,90	0,65	0,28

**Tabulka 3:** Vypočítaná verifikační kritéria

Verifikační kritéria nám říkají, že se modelu nejlépe daří na testovacím vzorku dat predikovat výskyt okluzních front s přesností o cca 10 procent vyšší než u studených a teplých front. Z hodnoty zkreslení můžeme usoudit, že vzhledem k její hodnotě, která leží v intervalu  $(0; 1)$ , je výskyt front podhodnocený a to v případě všech typů front. S tím se pojí viditelně nízký poměr falešných poplachů. Obecně lze říci, že přesnost modelu zatím není dostačující pro samostatné využití, ale zároveň není až tak nízká, aby byl model nepoužitelný. V současném stavu mohou generované linie front sloužit alespoň jako vodítko pro manuální tvorbu synoptických map.



## 9 Závěr

Cílem této práce bylo vytvořit koncept pro systém, který bude predikovat výskyt atmosférických front. Pro tento účel vznikl návrh fungování systému na bázi generativní adversariální sítě s ekosystémem opětovně využitelných utilit pro manipulaci s daty pro přípravu trénovacího datasetu. Celá práce je dostupná jako otevřený projekt, který umožňuje snadnější možnost navázání na toto dílo bez nutnosti začínat od počátku.

Data o úspěšnosti predikcí sice zatím nesvědčí o produkčním nasazení, ale poukazují na skutečnost, že tato cesta je správná a s malými modifikacemi a větším množstvím datových vstupů bude možné v brzké době fronty pomocí tohoto systému predikovat.

Díky kladnému ohlasu ve zpětné vazbě má tento projekt perspektivu a bude jej, včetně jeho dílčích částí, možné využít v dalších meteorologických aplikacích pro predikování nebezpečných jevů. To dle mého názoru potvrdilo stanovené cíle o dalším vývoji.

### 9.1 Budoucí rozvoj

Na tomto projektu jsou v blízké budoucnosti v plánu další vývojové práce, které se zaměří na optimalizaci neuronové sítě, drobné úpravy v oblasti extrakce původních obrázků synoptické situace, dopsání interfacu příkazové řádky u některých skriptů a hlavně na upravení formátu datasetů z obrázků na datová pole v NetCDF. To zaručí mnohem lepší škálovatelnost pro další meteorologické vstupy.

#### 9.1.1 Data

Jedno z limitací současného stavu je nedostatek situací. Jedním z řešení tohoto problému je použití jiných datových zdrojů, jako je například německá DWD a britský MetOffice. Cílem je tedy upravit extraktor synoptických map tak, aby byl plně nastavitelný konfiguračním souborem, který bude definovat hledané prvky, souřadnice oblastí pro zamaskování a kde hledat metadata.

Za účelem opravy časových posunů mezi daty z numerického modelu a synoptickou mapou bude vyvinut software pro identifikaci tohoto jevu a vybrání nejvhodnějšího časového bodu. To bude vyžadovat vyřešit optimalizaci přenosu velkého množství dat přes internet.

#### 9.1.2 Zdokonalení numerických produktů

Další z plánovaných prací je výzkum dalších fyzikálních prvků, které mají vliv na frontogenezi a jejich využití. Dalším krokem bude širší využití numerických metod k vymezení míst výskytu atmosférické fronty.

#### 9.1.3 Webová aplikace

S cílem otevřenosti kódu se pojí i otevřenost generovaných dat. Proto bude vytvořena webová aplikace, která bude sloužit jako repozitář aktuálních a minulých vygenerovaných situací pomocí tohoto systému. Jako součást je v plánu nástroj pro generování animací za určité období.

## 10 Bibliografie

- [1] *Meteorologie synoptická*. [online]. ČESKÁ METEOROLOGICKÁ SPOLEČNOST. Elektronický meteorologický slovník. ©2017  
Dostupné z: <http://slovník.cmes.cz/heslo/1990>. [cit. 2024-03-23].
- [2] *Hmota vzduchová*. [online]. ČESKÁ METEOROLOGICKÁ SPOLEČNOST. Elektronický meteorologický slovník. ©2017  
Dostupné z: <http://slovník.cmes.cz/heslo/1069>. [cit. 2024-03-23].
- [3] *Fronta atmosférická*. [online]. ČESKÁ METEOROLOGICKÁ SPOLEČNOST. Elektronický meteorologický slovník. ©2017  
Dostupné z: <http://slovník.cmes.cz/heslo/929>. [cit. 2024-03-23].
- [4] *Fronta studená*. [online]. ČESKÁ METEOROLOGICKÁ SPOLEČNOST. Elektronický meteorologický slovník. ©2017  
Dostupné z: <http://slovník.cmes.cz/heslo/946>. [cit. 2024-03-23].
- [5] VYSOUDIL, Miroslav. *Základy fyzické geografie 1: Meteorologie a klimatologie*. Univerzita Palackého v Olomouci. ©2013. ISBN 978-80-244-3893-1.
- [6] *Fronta teplá*. [online]. ČESKÁ METEOROLOGICKÁ SPOLEČNOST. Elektronický meteorologický slovník. ©2017  
Dostupné z: <http://slovník.cmes.cz/heslo/951>. [cit. 2024-03-23].
- [7] *Charakteristiky okluzní fronty*. [online]. METEO AKTUALITY – seriózní servis o počasí. ©2021 Dostupné z: <https://www.pocasimeteoaktuality.cz/charakteristiky-okluzni-fronty>. [cit. 2024-03-23].
- [8] *Frontogeneze*. [online]. ČESKÁ METEOROLOGICKÁ SPOLEČNOST. Elektronický meteorologický slovník. ©2017  
Dostupné z: <http://slovník.cmes.cz/heslo/959>. [cit. 2024-03-23].
- [9] *Teplotní frontální parametr*. [online]. ČESKÁ METEOROLOGICKÁ SPOLEČNOST. Elektronický meteorologický slovník. ©2017  
Dostupné z: <http://slovník.cmes.cz/heslo/2519>. [cit. 2024-03-27].
- [10] *Analýza synoptické situace*. [online]. In: ČESKÝ HYDROMETEOROLOGICKÝ ÚSTAV. Portál ČHMÚ. ©2024, 1.1.2024. Dostupné z: <https://www.chmi.cz/aktualni-situace/aktualni-stav-pocasi/evropa/synopticka-situace>. [cit. 2024-01-01].
- [11] BLUESTEIN, H.B. *Synoptic-dynamic Meteorology in Midlatitudes: Observations and theory of weather systems*. Oxford University Press, ©1993. ISBN 9780195062687.
- [12] *Předpovědní synoptická mapa Evropy*. [online]. In: ČESKÝ HYDROMETEOROLOGICKÝ ÚSTAV. Portál ČHMÚ. ©2024, 24.3.2024. Dostupné z: <https://www.chmi.cz/predpovedi/predpovedi-pocasi/evropa/synopticka-situace>. [cit. 2024-03-24].
- [13] *Frontogenesis*. [online]. MetPy 1.6dev ©2024  
Dostupné z: <https://unidata.github.io/MetPy/latest/api/generated/metpy.calc.frontogenesis.html>. [cit. 2024-03-23].

- [14] RENARD RJ, CLARKE LC. ©1965 *Experiments in numerical objective frontal analysis*. *Monthly Weather Review* 93: 547–556. DOI: 10.1175/1520-0493(1965)093;0547:EINOFA;2.3.CO;2.
- [15] FUNK, Ted ©2011 *A practical guide to quasi-gastrophic theory, response to gastrophic deformation, agestrophic deformation, atmospheric motion, jet streaks*. [online]. Dostupné z: [https://www.weather.gov/media/lmk/soo/QG\\_Theory\\_Review.pdf](https://www.weather.gov/media/lmk/soo/QG_Theory_Review.pdf). [cit. 2024-03-28].
- [16] *ECMWF Reanalysis v5 (ERA5)* [online]. EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS. ECMWF Reanalysis v5. ©2024  
Dostupné z: <https://www.ecmwf.int/en/forecasts/dataset/ecmwf-reanalysis-v5>. [cit. 2024-03-27].
- [17] AMAZON WEB SERVICES. *What is Python? - Python Programming Language Explained*. [online]. ©2024  
Dostupné z: <https://aws.amazon.com/what-is/python/>. [cit. 2024-03-28].
- [18] META AI. *PyTorch*. [online]. ©2024  
Dostupné z: <https://ai.meta.com/tools/pytorch/>. [cit. 2024-03-28].
- [19] *Introduction*. [online]. OpenCV documentation ©2024  
Dostupné z: <https://docs.opencv.org/4.x/d1/dfb/intro.html>. [cit. 2024-03-29].
- [20] *GDAL: Geospatial Data Abstraction Library*. [online]. PyPI ©2024  
Dostupné z: <https://pypi.org/project/GDAL/>. [cit. 2024-03-29].
- [21] *GTK*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2023.  
Dostupné z: <https://cs.wikipedia.org/wiki/GTK>. [cit. 2024-03-29].
- [22] *GRIB: General Regularly-distributed Information in Binary form*. [online]. DEUTSCHER WETTERDIENST. Information technology. ©2024  
Dostupné z: <https://www.dwd.de/EN/aboutus/it/functions/Teasergroup/grib.html>. [cit. 2024-03-29].
- [23] TIŠNOVSKÝ, Pavel. Framework Torch: základy práce s neuronovými sítěmi. *Root.cz* [online]. ©2017 [cit. 2024-03-30]. Dostupné z: <https://www.root.cz/clanky/framework-torch-zaklady-prace-s-neuronovymi-sitemi>
- [24] VOLNÁ, Eva. *Neuronové sítě 1*. [online]. Druhé. Ostravská univerzita v Ostravě, 2008. Dostupné z: [https://web.osu.cz/~Volna/Neuronove\\_site\\_skripta.pdf](https://web.osu.cz/~Volna/Neuronove_site_skripta.pdf). [cit. 2024-03-30].
- [25] ISOLA Phillip, ZHU Jun-Yan, ZHOU Tinghui, EFROS Alexei A. ©2017 *Image-to-Image Translation with Conditional Adversarial Networks*. DOI: 10.48550/arXiv.1611.07004.
- [26] TOMAR, Nikhil. What is UNET? *Medium.com* [online]. ©2021. Dostupné z: <https://medium.com/analytics-vidhya/what-is-unet-157314c87634> [cit. 2024-03-30]

- [27] *U-Net: Convolutional Networks for Biomedical Image Segmentation*. [online]. University of Freiburg. ©2015. Dostupné z: <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net>. [cit. 2024-03-30].
- [28] NAGORNY, Pierre. *Generative adversarial networks for geometric surfaces prediction in injection molding: Performance analysis with Discrete Modal Decomposition*. IEEE, ©2018.
- [29] ALJOHANI A, ALHARBE N. *Generating Synthetic Images for Healthcare with Novel Deep Pix2Pix GAN*. Online. ©2022; DOI: 10.3390/electronics11213470.
- [30] ŠAUR, David. *Hodnocení úspěšnosti předpovědí*. Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně, ©2020.

## Seznam obrázků

1	Mapa analýzy synoptické situace z 1. 1. 2024 12:00 UTC[10]	7
2	Velikost gradientu potenciální teploty	9
3	Termální frontální parametr v hladině 850 hPa	10
4	Rychlost proudění větru v hladině 300 hPa	11
5	Vypočtená frontogeneze z 23. 3. 2024 12:00 UTC	12
6	Předpovědní synoptická mapa na 23. 3. 2024 12:00 UTC[12]	12
7	Srovnání frontogeneze nad Grónskem s ledovou pokrývkou	13
8	Frontogeneze s aplikovanou maskou	13
9	Směry Q-vektorů na teplotě v 850 hPa	14
10	Q-vektory zobrazené šipkami a jejich divergence	14
11	Relativní vorticity v hladině 850 hPa	15
12	Analýza synoptické situace z 18. 5. 2020 [10]	15
13	Schéma architektury softwaru	21
14	Kroky při extrakci front	25
15	Vygenerovaný pár obrázků pro trénink	30
16	GUI aplikace Dataset inspector	30
17	Zjednodušený model neuronu	31
18	Architektura generátoru [29]	33
19	Problematická predikce	34
20	Srovnání po úpravě ztrátové funkce	35
21	Srovnání průběhů ztrát při tréninku	36

## Seznam ukázek kódu

1	Třída definující konvoluční blok	18
2	Tvorba masek front	18
3	Skript shellu pro augmentaci datasetu	19
4	Zjednodušená funkce pro extrakci linií z masky fronty	24
5	Skript pro georeferenci pomocí GDAL	26
6	Ukázka GeoJSONu	26
7	Práce s datasetem pomocí xarray a MetPy	28
8	Programové znázornění výpočtu Hausdorffovy ztráty	35

## Seznam tabulek

1	Přehled verifikačních kategorií	37
2	Verifikační kritéria predikce	38
3	Vypočítaná verifikační kritéria	38

# Přílohy

1. Zdrojový kód – online přístupný na <https://github.com/pavelpernicka/frontanim>
2. Odborný posudek – Ing. David Šaur, Ph.D.

## ODBORNÝ POSUDEK K PRÁCI SOČ

Navržený systém pro predikci výskytu atmosférických front lze využít pro vědecko-výzkumné účely zaměřené např. pro zpřesnění předpovědi místa výskytu konvektivních bouří a jejich nebezpečných dopadů jako přívalových srážek spojených s krupobitím a silnými nárazy větru. Výstupy tohoto systému by byly použitelné např. v již realizovaném výzkumném projektu č. *VI20192022134* – „*Systém zpřesněné předpovědi konvektivních srážek pro krajský územní celek*“, kde tyto metody by bylo možné aplikovat pro pokročilé metody rozpoznávání obrazu s využitím neuronových sítí pro výběr předpovídané nebo aktuální skupiny situací nebo jedné situace tzv. drah bouří s relativní četností výskytu konvektivních srážek pro území České republiky v rámci tzv. statistického odhadu aktuálního nebo budoucího výskytu bouřek. Metoda drah bouří má zásadní význam pro zpřesněnou předpověď a budou využity i v rámci plánovaného výzkumného projektu „*TORPIS – Systém včasného varování před nebezpečím výskytu tornád pro krajský územní celek*“ v rámci budoucích projektových výzkumných tuzemských nebo zahraničních výzev. Výstupy tohoto systému v rámci tohoto projektu mohou zpřesnit statistický odhad podmínek pro výskyt supercelárních tornadických bouří na základě dostupných synoptických předpovědí na území ČR.

V rovině akademické a vědecko-výzkumné, aktuální a budoucí vývoj hlavních výsledků projektu č. *VI20192022134* – „*Systém zpřesněné předpovědi konvektivních srážek pro krajský územní celek*“ je řešen ve spolupráci s Univerzitou Tomáše Bati ve Zlíně, Fakultou aplikované informatiky, Ústavem umělé inteligence a informatiky, konkrétně s vědeckým pracovníkem Ing. Tomášem Turečkem s využitím vybraných konvolučních neuronových sítí. Spolupráce v oblasti meteorologie probíhá s doc. Mgr. Michalem Žákem, Ph.D., akademickým pracovníkem na Univerzitě Karlově, Matematicko-fyzikální fakulty v oboru Meteorologie a klimatologie s cílem zpřesnit predikce polohy front s využitím dalších a přesnějších meteorologických parametrů detailněji popisující proces frontogeneze v atmosféře.

V rovině implementace výsledků výzkumu do praxe, bude v následujících letech tato metoda využita pro zpřesnění statistického odhadu konvektivních srážek v aplikaci SSPKS KUC – „*Systém statistické předpovědi konvektivních srážek pro krajský územní celek*“. Tato aplikace je právě implementována do krajské ICT infrastruktury Zlínského kraje pro účely včasného varování před nebezpečím silných konvektivních a přívalových srážek pro krajské a obecní krizové a povodňové orgány. Mimo jiné, výstup „*Předpověď přívalových srážek*“ z aplikace SSPKS KUC je využíván pro aktuální odhad a předpověď nebezpečí přívalových povodní v aplikaci FLAPRIS – „*Systém pro podporu zpřesněné a včasné předpovědi nebezpečí vzniku přívalových povodní a usnadnění činností krizových a povodňových orgánů kraje*“, již ukončeného projektu č. VB01000008. Tato aplikace aktuálně běží již v ostrém provozu a zájem o ni projevilo Krajské ředitelství Hasičského záchranného sboru. Ve všech těchto aplikacích lze využít efektivně navrženou aplikaci Frontamin pro účely zvyšování přesnosti a spolehlivosti výstupů pro předpověď silných konvektivních bouří, které mohou produkovat příčinné přívalové srážky vedoucích ke vzniku lokálních přívalových povodní nejen pro území Zlínského kraje, ale v rámci budoucích projektů i pro testování na Slovensku, př. v dalších evropských státech.

Ve Zlíně dne 10.05.2024

  
.....  
Ing. David Šaur, Ph.D.