

**UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH  
PRÍRODOVEDECKÁ FAKULTA**

**PREDIKCIA OBLAČNOSTI Z RADAROVÝCH DÁT  
POMOCOU NEURÓNOVÝCH SIETÍ**

**Bakalárska práca**

**2024**

**Bianka Szepesiová**

**UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH  
PRÍRODOVEDECKÁ FAKULTA**

**PREDIKCIA OBLAČNOSTI Z RADAROVÝCH DÁT  
POMOCOU NEURÓNOVÝCH SIETÍ**

**Bakalárska práca**

Študijný program: Informatika  
Študijný odbor: 9.2.1. Informatika  
Školiace pracovisko: Ústav informatiky  
Vedúci práce: RNDr. Richard Staňa

**Košice 2024**

**Bianka Szepesiová**



## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Bianka Szepesiová

**Študijný program:** informatika (jednoodborové štúdium, bakalársky I. st., denná forma)

**Študijný odbor:** Informatika

**Typ záverečnej práce:** Bakalárska práca

**Jazyk záverečnej práce:** slovenský

**Sekundárny jazyk:** anglický

**Názov:** Predikcia oblačnosti z radarových dát pomocou neurónových sietí

**Názov EN:** Cloud forecasting from radar data using neural networks

**Ciel:**  
1. Analyzovať existujúce metódy predikcie oblačnosti.  
2. Implementovať vlastný model neurónovej siete na predikciu oblačnosti.  
3. Analyzovať dosiahnuté výsledky pomocou rôznych metrík.

**Literatúra:**  
1. Fang, W., Xue, Q., Shen, L., & Sheng, V. S. (2021). Survey on the application of deep learning in extreme weather prediction. *Atmosphere*, 12(6), 661.  
2. Shi, E., Li, Q., Gu, D., & Zhao, Z. (2017). Convolutional neural networks applied on weather radar echo extrapolation. *DEStech Trans. Comput. Sci. Eng.*  
3. Berthomier, L., Pradel, B., & Perez, L. (2020, November). Cloud cover nowcasting with deep learning. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)* (pp. 1-6). IEEE.  
4. SHMÚ. Slovenská rádiolokačná siet. Dostupné na internete: <https://www.shmu.sk/sk/?page=1566>, 2024.

**Vedúci:** RNDr. Richard Staňa

**Oponent:** doc. RNDr. Ľubomír Antoni, PhD.

**Ústav :** ÚINF - Ústav informatiky

**Riaditeľ ústavu:** doc. RNDr. Ondrej Krídlo, PhD.

**Dátum schválenia:** 17.05.2024

## **Pod'akovanie**

Týmto by som chcela pod'akovat' RNDr. Richardovi Staňovi za jeho ochotu, trpezlivosť a čas, ktorý mi venoval počas vypracovania mojej bakalárskej práce. Ďalej by som rada pod'akovala Slovenskému meteorologickému ústavu za poskytnutie datasetu, ktorý bol pre túto prácu klúčovým.

## Abstrakt

Táto bakalárska práca sa venuje problematike predikcie oblačnosti s využitím rada-rových dát a moderných techník neurónových sietí. Súčasný stav meteorológie vyžaduje presnejšie a sofistikovanejšie nástroje na predpoved' počasia, a práve v tomto kontexte sa stávajú neurónové siete neoceniteľnými nástrojmi pre analýzu a predpoved' kom-plexných meteorologických javov. Cieľom práce je skúmať možnosti využitia neurónových sietí na predikciu oblačnosti. Teoretická časť práce sa zameriava na prehľad existujúcich metód predikcie oblačnosti, s dôrazom na ich výhody a obmedzenia. V praktickej časti navrhujeme a implementujeme vlastný model neurónovej siete, ktorý je testovaný na dátach poskytnutých Slovenským hydrometeorologickým ústavom. V záverečnej časti práce rozoberáme dosiahnuté výsledky a navrhujeme možnosti vy-lepšenia do budúcnosti.

**Kľúčové slová:** *Predikcia oblačnosti, Radarové dátá, Neurónové siete, ConvLSTM, Enkóder, Dekodér.*

## Abstract

This bachelor's thesis addresses the challenge of cloud prediction using radar data and modern neural network techniques. The current state of meteorology requires more precise and sophisticated tools for weather forecasting, and within this context, neural networks prove to be invaluable tools for analysing and predicting complex meteorological phenomena. The aim of this thesis is to explore the potential of utilising neural networks for cloud prediction. The theoretical part of the thesis focuses on an overview of existing methods for cloud prediction, emphasising their advantages and limitations. In the practical part, we propose and implement our own neural network model, which is tested on data provided by the Slovak Hydrometeorological Institute. In the final part of the thesis, we discuss the achieved results and suggest possible improvements for the future.

**Keywords:** *Cloud prediction, Radar data, Neural networks, ConvLSTM, Encoder, Decoder.*

# Obsah

<b>Úvod</b>	<b>8</b>
<b>1 Prehľad existujúcich metód predikcie oblačnosti</b>	<b>9</b>
1.1 Tradičné metódy predikcie oblačnosti . . . . .	9
1.1.1 Satelitné metódy . . . . .	9
1.1.2 Radarové metódy . . . . .	10
1.1.3 Numerické prediktívne metódy . . . . .	10
1.2 Neurónové siete v kontexte predikcie oblačnosti . . . . .	11
1.2.1 Rekurentná dynamická konvolučná neurónová sieť . . . . .	11
1.2.2 U-Net . . . . .	12
1.2.3 RainPredRNN . . . . .	13
<b>2 Dataset</b>	<b>15</b>
2.1 Získanie dát . . . . .	15
2.2 Spracovanie dát . . . . .	15
<b>3 Návrh a implementácia vlastného riešenia</b>	<b>17</b>
3.1 ConvLSTM . . . . .	17
3.1.1 Architektúra . . . . .	17
3.1.2 Tréning . . . . .	20
3.2 Rozšírenie ConvLSTM o enkóder a dekodér . . . . .	22
3.2.1 Architektúra . . . . .	22
3.2.2 Tréning . . . . .	25
<b>4 Výsledky</b>	<b>29</b>
4.1 ConvLSTM . . . . .	29
4.2 Rozšírenie ConvLSTM o enkóder a dekodér . . . . .	35
<b>Záver</b>	<b>41</b>
<b>Zoznam použitej literatúry</b>	<b>45</b>



# Úvod

Predikcia oblačnosti je neodmysliteľnou súčasťou predpovede počasia a ovplyvňuje širokú škálu faktorov, vrátane teploty, zrážok a intenzity slnečného žiarenia. Okrem toho, že má vplyv na bežné aktivity ako výlety a športové udalosti, má tiež významný dopad na mnohé ďalšie oblasti, napríklad na poľnohospodárstvo a dopravu. Rovnako dôležitú úlohu zohráva aj v energetickom sektore, kde má podstatný vplyv na výkon solárnych systémov a výrobu energie z obnoviteľných zdrojov.

V súčasnosti sa pri predikcii oblačnosti často spoliehame na tradičné metódy, ako sú satelitné a radarové techniky a numerické prediktívne modely. Napriek ich užitočnosti majú tieto metódy svoje obmedzenia, najmä pokial ide o predikciu oblačnosti v krátkodobom časovom horizonte. S nástupom moderných technológií, ako sú neurónové siete, však vzniká nová perspektíva v oblasti meteorológie. Neurónové siete majú schopnosť efektívne analyzovať časové rady. Okrem toho sú široko využívaným nástrojom v oblasti spracovania obrazu, čím sa stávajú ideálnym nástrojom pre predikciu oblačnosti. Táto kombinácia prináša obrovský potenciál vylepšiť presnosť predikcií a získať nové poznatky o dynamike počasia.

Cieľom tejto bakalárskej práce je preskúmať a analyzovať existujúce metódy predikcie oblačnosti a následne navrhnúť a implementovať vlastný model neurónovej siete na predikciu oblačnosti z radarových dát. Ďalej sa zameriame na analýzu dosiahnutých výsledkov pomocou rôznych metrík, čím zhodnotíme efektívnosť navrhnutého modelu.

V kapitole 1 sa budeme venovať analýze tradičných metód predikcie oblačnosti, vrátane ich výhod a obmedzení, ako aj podrobnej analýze použitia neurónových sietí v kontexte predikcie oblačnosti, s dôrazom na ich architektúru a výkonnosť. V kapitole 2 budeme diskutovať o procese získavania a spracovania dát, vrátane techník a nástrojov použitých na ich úpravu. Následne sa v kapitole 3 budeme venovať návrhu a implementácii vlastného riešenia založeného na konvolučnej LSTM (Long Short Term Memory) neurónovej sieti a jej rozšírení o Enkóder a Dekódér architektúru. V kapitole 4 predstavíme a analyzujeme dosiahnuté výsledky natrénovaných modelov a porovnávame ich pomocou rôznych metrík.

# 1 Prehľad existujúcich metód predikcie oblačnosti

V tejto kapitole preskúmame tradičné prístupy k predpovedi oblačnosti, ako aj moderné technologické inovácie, ktoré ovplyvnili oblasť meteorologickej predikcie. Zameriame sa na rôzne aspekty predikcie oblačnosti, od tradičných meteorologickejch metód po využitie moderných neurónových sietí. Prechádzame cez výhody a obmedzenia každej metódy s cieľom poskytnúť celkový pohľad na súčasný stav výskumu a praxe v oblasti predpovede oblačnosti.

## 1.1 Tradičné metódy predikcie oblačnosti

### 1.1.1 Satelitné metódy

Satelitné metódy predpovede počasia využívajú dátu z meteorologickejch druzíc, ktoré snímajú elektromagnetické žiarenie v rôznych spektrálnych oblastiach. Tieto dátu umožňujú sledovať oblačnosť, teplotu povrchu, pohyb oblakov, detekciu zrážok a meranie vlhkosti v atmosfére. Vďaka infračerveným snímkam je možné efektívne monitorovať teplotné rozdiely, hustotu mrakov a intenzitu zrážok aj v nočných hodinách. Pomocou snímkov vodnej pary je ľahké monitorovať smer a rýchlosť vetra, aj keď sa na viditeľných a infračervených snímkach zdá obloha jasná. Celkovo sú satelitné metódy neoddeliteľnou súčasťou výskumu, sledovania a predpovede počasia na celosvetovej úrovni.

Satelitné metódy sú výhodné pre svoj globálny pohľad na atmosféru a schopnosť poskytovať rýchle aktualizácie v reálnom čase, čo je obzvlášť užitočné pri sledovaní rozsiahlych meteorologickejch vzorov. Napriek tomu majú svoje obmedzenia, ako napríklad nižšie rozlíšenie. [15]

### **1.1.2 Radarové metódy**

Radarové metódy predikcie počasia využívajú radarové systémy na získanie informácií o aktuálnych meteorologických podmienkach. Tieto systémy emitujú rádiové vlny do atmosféry, ktoré sa odrazia od zrážok, ako sú dažďové kvapky alebo snehové vločky. Odrazené rádiové vlny sú zachytené radarovou anténou a spracované na vytvorenie radarového obrazu. Tento obraz zobrazuje intenzitu a polohu zrážok v reálnom čase.

Meteorológovia potom analyzujú tieto dát, čo umožňuje predpovedať pohyb a intenzitu zrážok v krátkodobom horizonte. Radarové informácie sa často kombinujú s ďalšími zdrojmi dát, ako sú satelitné snímky a pozemné merania, čo viedie k presnejším predpovediam počasia. Celý proces umožňuje sledovanie vývoja atmosférických podmienok a poskytuje dôležité informácie pre predpoved počasia s vysokou presnosťou.

Radarové metódy vynikajú vysokým rozlíšením a schopnosťou okamžite detegovať zrážky, čím sú ideálnymi pre detailné monitorovanie lokálnych poveternostných podmienok. Ich obmedzenia sa týkajú hlavne dosahu a neschopnosti zaznamenávať signály v ich tesnej blízkosti. Ďalším obmedzením je, že radar môže niekedy zachytiť slabé ozveny z iných objektov, ako napríklad lietadlá, dym z veľkých požiarov alebo kŕdle vtákov. [8]

### **1.1.3 Numerické prediktívne metódy**

Numerické metódy predikcie počasia využívajú matematické modely atmosféry. Tieto modely aplikujú fyzikálne zákony a rovnice na popis zmeny meteorologických premenných v priestore a čase. Celý proces začína s inicializáciou, kde sa určia počiatocné podmienky atmosféry na základe reálnych meteorologických pozorovaní. Následne sa atmosféra rozdelí na mriežku a model rieši rovnice pre každý bod mriežky v čase. Numerické metódy, ako metóda konečných diferencií, potom aproximujú zmeny v priestore a čase. Výpočtová kapacita je klíčovým faktorom pre efektívne výpočty a moderné superpočítače umožňujú vysokú presnosť modelov. Výsledky predpovedí sú následne verifikované porovnaním s aktuálnymi meteorologickými pozorovaniami. Ak sa predpovede líšia, model môže byť aktualizovaný.

Numerické metódy umožňujú široké pokrytie a modelovanie komplexných atmosférických procesov, ale majú vysokú výpočtovú náročnosť, sú závislé na presných počiatocných podmienkach a potrebujú neustálu kalibráciu a aktualizáciu. Ich flexibilita však umožňuje prispôsobenie sa rôznym geografickým oblastiam a časovým horizontom. [6]

## 1.2 Neurónové siete v kontexte predikcie oblačnosti

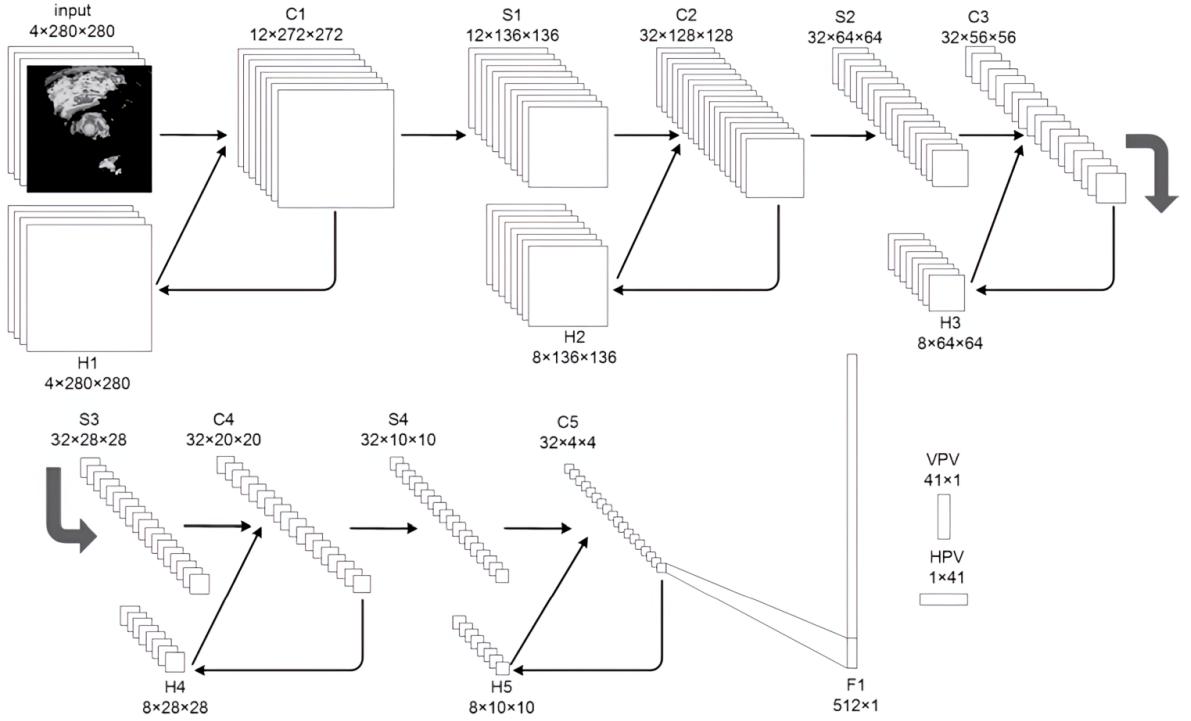
### 1.2.1 Rekurentná dynamická konvolučná neurónová sieť

V článku [12] je predstavený navrhnutý model RDCNN (Recurrent Dynamic CNN), ktorý využíva konvolučné neurónové siete na extrapoláciu radarových dát v predpovedi počasia. Model RDCNN sa skladá z dvoch hlavných častí: rekurentnej dynamickej podsiete (RDSN) a vrstvy pravdepodobnostnej predikcie (PPL). Ukážka siete RDSN použitej v tomto článku je na obrázku 1.

**Rekurentná dynamická podsiet' (RDSN)** zahŕňa konvolučné vrstvy, ktoré postupne spracúvajú vstupné obrázky a extrahujú rôzne príznaky pomocou konvolúcií. Vzorkovacie vrstvy slúžia na znižovanie rozlíšenia výstupných obrázkov, zatiaľ čo skryté vrstvy vytvárajú cyklickú štruktúru medzi konvolučnými vrstvami a uchovávajú historické informácie z trénovacieho procesu. Klasifikátor generuje dva pravdepodobnostné vektory: VPV (vertikálna predikcia) a HPV (horizontálna predikcia), pričom sú tieto vektory vypočítané pomocou softmax funkcie na základe výstupu klasifikátora.

**Vrstva pravdepodobnostnej predikcie (PPL)**, ktorá nasleduje po RDSN, má na vstupe posledný obrázok v sekvenции a pravdepodobnostné vektory VPV a HPV z RDSN. Skladá sa z dvoch konvolučných vrstiev, ktoré používajú pravdepodobnostné vektory ako konvolučné jadrá na predikciu ďalšieho radarového obrázka.

V rámci experimentov a vyhodnotenia bol model RDCNN podrobnený testovaniu na troch nezávislých datasetoch, získaných z radarových dát v troch mestách (Nanjing, Hangzhou, Xiamen) v roku 2016. Výsledky RDCNN boli porovnávané s tradičnými metódami, najmä s COTREC. Hodnotenie modelu zahrňovalo aj predpovede počasia v troch zrážkových udalostiach, pričom RDCNN dosiahol presnejšie predpovede v porovnaní s alternatívnymi metódami. Tieto experimenty a výsledky podporujú účinnosť a presnosť modelu RDCNN pri predikcii radarových dát a jeho potenciál pre využitie v oblasti predpovede počasia.



Obr. 1: Celková štruktúra RDSN s piatimi konvolučnými vrstvami ( $C_1, C_2, C_3, C_4, C_5$ ), štyrmi vzorkovacími vrstvami ( $S_1, S_2, S_3, S_4$ ), piatimi skrytými vrstvami ( $H_1, H_2, H_3, H_4, H_5$ ) a jedným klasifikátorom. [12]

### 1.2.2 U-Net

V článku [1] sa autori z AI Lab METEO FRANCE zaoberejú výzvou predikcie oblačnosti na krátku dobu, konkrétnie na niekoľko hodín dopredu. Hlavným cieľom autorov je predikovať pohyb oblakov a zlepšiť presnosť predpovedí v krátkodobom horizonte. Na tento účel využívajú hlboké konvolučné siete (CNNs), s dôrazom na konkrétnu architektúru U-Net. Táto sieť je aplikovaná na satelitné snímky série družíc METEOSAT a slúži na segmentáciu obrazu a predikciu pohybu oblakov.

Výsledky experimentov ukazujú, že modely založené na hlbokom učení, predovšetkým U-Net, dosahujú významne lepšie výsledky v porovnaní s tradičnými meteorologickými metódami. Konkrétnie prekonávajú perzistencia, ktorá predstavuje využitie posledného pozorovania ako predpovede pre každý časový krok. Tieto modely sa osvedčujú v predikcii pohybu oblakov, pričom dosahujú presnejšie výsledky aj v porovnaní s modelom AROME, ktorý využíva fyzikálne modelovanie.

Zaujímavým výsledkom je skutočnosť, že aj jednoduchý model s jedným vstupom (obrázkom) dokázal prekonať perzistencia v predikcii pohybu oblakov. To naznačuje, že aj s minimálnym množstvom informácií môže hlboké učenie poskytovať významné

zlepšenia v predikcii meteorologických javov.

Autori tvrdia, že hlboké učenie, konkrétnie U-Net architektúra, môže byť efektívnym nástrojom pre predpovedanie pohybu oblakov v krátkodobom horizonte, s potenciálom poskytnúť presnejšie výsledky v porovnaní s tradičnými metódami.

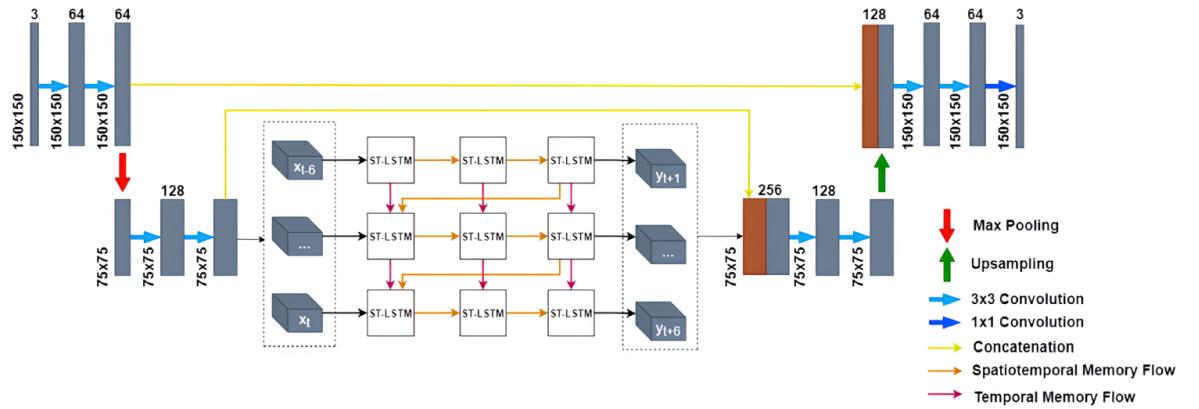
### 1.2.3 RainPredRNN

V článku [16] Tuyen a spol. predstavujú nový model RainPredRNN, ktorý kombinuje vlastnosti U-Net a PredRNN\_v2 s cieľom optimalizovať časovú náročnosť trénovania a testovania, pričom model zachováva komplexné priestorové vlastnosti radarových dát.

Autori diskutujú o prínosoch ST-LSTM (Spatiotemporal Long Short-Term Memory) neurónovej siete, ktorá je základom prediktívnej rekurentnej neurónovej siete (PredRNN). Wang a spol., ktorí PredRNN navrhli, si všimli, že pamäťové stavy sa automaticky nerozlišujú a dochádza k ich spontánnemu odpojeniu. Na základe toho navrhli novú stratovú funkciu s názvom „memory decoupling“, ktorá kombinuje štandardnú stratovú funkciu strednej kvadratickej chyby s novou funkciou regularizácie odpojenia pamäte a vytvorili tak novú verziu tohto modelu s názvom PredRNN\_v2 [18].

RainPredRNN využíva kľúčové vlastnosti U-Net modelu, ako sú enkóder-dekodér cesty a technika konkatenácie. Pôvodné obrázky sú spracované pomocou enkóder cesty, ktorá zmenšuje priestorové rozmery vstupov pred ich odoslaním do ST-LSTM vrstiev, čo výrazne znižuje výpočtovú náročnosť. Následne je pridaná dekodér cesta za ST-LSTM vrstvami, kde je využitá technika „skip-connections“ na zachovanie informácií a predchádzanie problému zanikania gradientov. Táto modifikácia výrazne zlepšuje efektivitu trénovania a testovania bez straty výkonnosti modelu. Ukážka tohto modelu je na obrázku 2.

Experimenty na reálnych dátach z radarovej stanice Phadin v provincii Dien Bien vo Vietname preukázali, že RainPredRNN výrazne skracuje čas trénovania a testovania, pričom stále poskytuje vysokú kvalitu predpovedaných obrázkov. V porovnaní s predchádzajúcimi modelmi, PredRNN a PredRNN\_v2, dosahuje RainPredRNN podobné alebo dokonca lepšie výsledky pri výrazne nižších časových a výpočtových nákladoch.



Obr. 2: Model RainPredRNN. Boxy s textom ST-LSTM označujú konvenčné priestorovo-časové LSTM, sivé boxy reprezentujú obrázky v rôznych úrovniach spracovania a hnedé boxy sú kópie orezaných príznakových máp. [16]

# 2 Dataset

## 2.1 Získanie dát

Dáta použité v tejto bakalárskej práci boli poskytnuté Slovenským hydrometeorologickým ústavom (SHMÚ). Tieto dáta predstavujú sériu radarových obrazov z obdobia od 1. januára 2016 do 30. septembra 2023, s frekvenciou každých 5 minút. Počty zahrnutých radarov sa líšia v závislosti od aktuálnej situácie vzhľadom na poruchy. V prvom polroku 2016 boli v prevádzke iba 2 radary, pričom neskôr pribudli nové zariadenia.

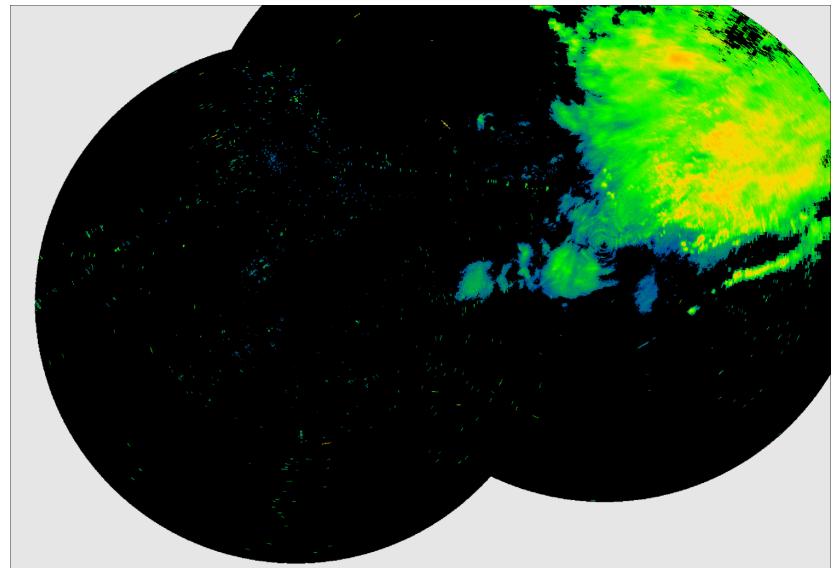
Štyri moderné meteorologické radary na Slovensku, typu Meteor 735 CDP10, sú umiestnené na Malom Javorníku, Kojšovskej holi, Kubínskej holi a Špaňom laze. Tieto radary poskytujú údaje o oblačnosti a zrážkach každých 5 minút. Výsledky sú spracúvané na centrálnom serveri SHMÚ Koliba a verejne dostupné na webovej stránke SHMÚ. [14]

## 2.2 Spracovanie dát

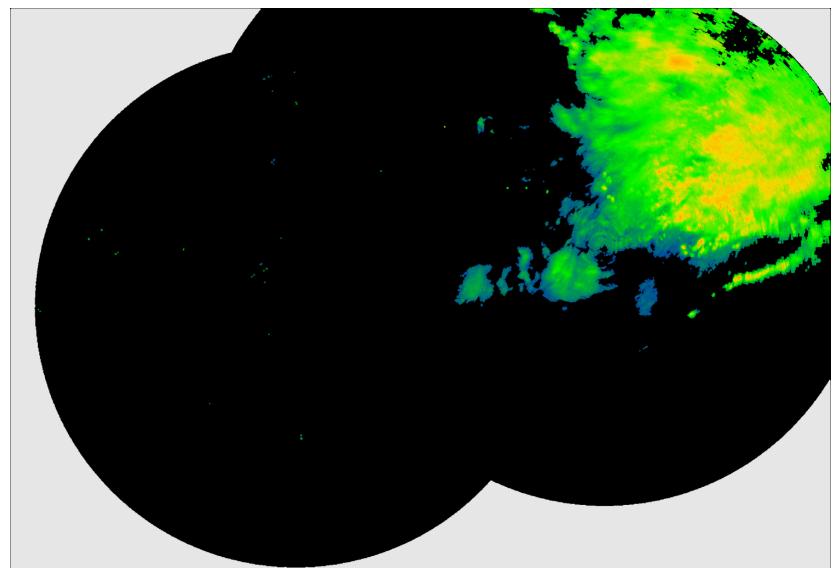
Na spracovanie dát sme použili niekoľko operácií s využitím knižnice OpenCV [4] na úpravu obrázkov. Cieľom týchto operácií bolo odstránenie šumu z obrázkov a zároveň zmenšenie ich rozlíšenia s cieľom zefektívniť proces trénovania modelov.

Najprv sme na pôvodný obrázok aplikovali morfológickú operáciu otvorenia, čo zahŕňa eróziu nasledovanú dilatáciou. Tento postup pomáha prerušiť tenké spoje medzi objektmi a odstrániť šum, čím dochádza k vyhľadeniu obrázka a eliminácii niektorých nepotrebných detailov.

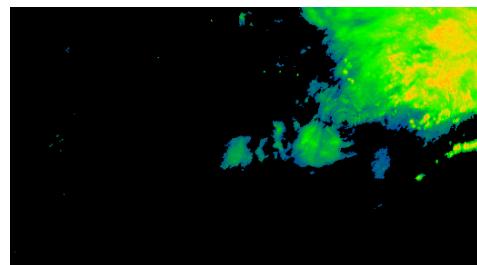
Pokračovali sme orezaním a zmenšením veľkosti obrázka. Špecificky sme vybrali oblasť obrázka, ktorá obsahovala hlavne Slovensko a následne sme túto oblasť zmenšili na rozmer 517x288 pixelov. Tento krok bol významný pre redukciu počtu výpočtových operácií počas trénovania modelov, keďže menšie obrázky vyžadujú menej výpočtových zdrojov. Na obrázkoch 3, 4 a 5 je možné vidieť postup transformácie dát.



Obr. 3: Pôvodný obrázok.



Obr. 4: Obrázok po morfológických úpravách.



Obr. 5: Orezaný a zmenšený obrázok.

# 3 Návrh a implementácia vlastného riešenia

Pre implementáciu sme zvolili programovací jazyk Python a na vytvorenie modelu sme použili knižnicu PyTorch [10]. Výstupné obrázky vo formáte GIF sme generovali pomocou knižnice Pillow [5]. Trénovanie modelu prebiehalo na serveri s procesorom AMD EPYC 7543P, 512 GB RAM a 2x NVIDIA A100 s pamäťou 40 GB, pričom pre trénovanie sme využívali obe dostupné GPU. Pri vyhodnocovaní výsledkov sme využívali nástroj Weights & Biases [2]. Implementácia je k dispozícii v našom repozitári na GitHub, kde sme zverejnili zdrojový kód a ďalšie relevantné materiály, vrátane generovaných GIF súborov. Vzhľadom na ich veľkosť sme previedli tieto GIF súbory do formátu MP4. Odkaz na GitHub sa nachádza v prílohe A.

## 3.1 ConvLSTM

### 3.1.1 Architektúra

Tento model je zložený z troch komponentov: ConvLSTMCell, ConvLSTM a Seq2Seq, ktoré spolu tvoria architektúru konvolučnej LSTM siete.

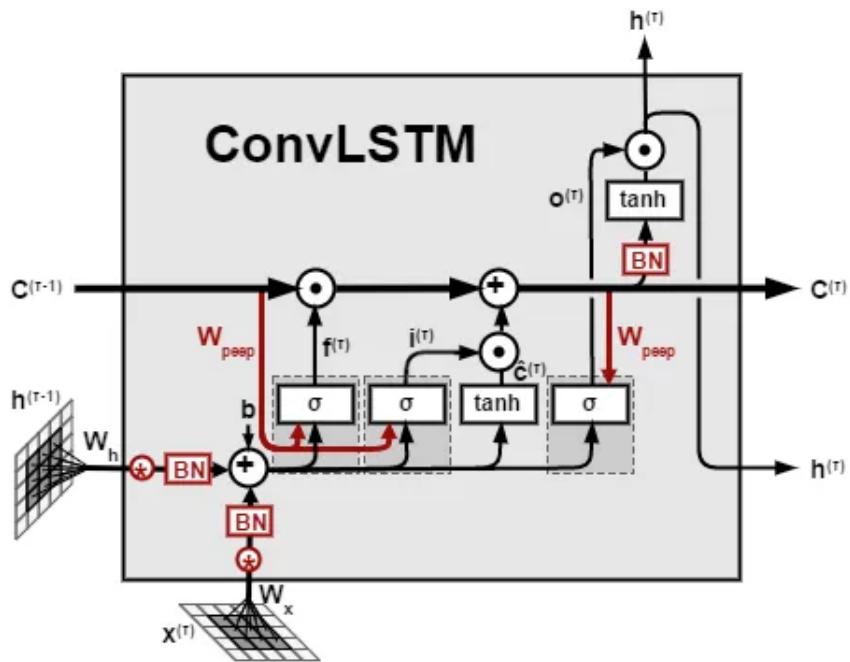
**ConvLSTMCell:** Táto bunka predstavuje základnú stavebnú jednotku modelu. Pozostáva z dvoch častí: konvolučnej a LSTM.

V prvom kroku sa vykonáva konvolúcia, pričom počet vstupných kanálov je rovný súčtu kanálov vstupného obrázka (alebo obrázka z predchádzajúcej vrstvy) a obrázka z predchádzajúcej pamäťovej bunky. Výstupom tejto konvolúcie je obrázok s počtom kanálov, ktorý je štyrikrát väčší ako počet kanálov obrázka z predchádzajúcej vrstvy.

Tento výstupný obrázok slúži ako vstup do LSTM bunky, pričom jeho kanály sú rozdelené na štyri rovnaké časti. Tieto časti sú potom použité na výpočet jednotlivých brán (vstupná brána, brána zabudnutia, výstupná brána) a nového stavu pamäťovej bunky.

Vstupná brána a brána zabudnutia sú vypočítané pomocou aktivačnej funkcie Sig-

moid, pričom sa používa Hadamardov súčin medzi príslušnými váhami a predchádzajúcim stavom pamäťovej bunky. Pomocou kombinácie aktuálneho stavu (súčin vstupnej brány a aktuálneho výstupu konvolúcie, na ktorý je aplikovaná aktivačná funkcia ReLU alebo tanh) a zabudnutého stavu (súčin brány zabudnutia a predchádzajúceho stavu bunky) dostaneme nový stav pamäťovej bunky. Výstupná brána je tiež vypočítaná pomocou aktivačnej funkcie Sigmoid a Hadamardoveho súčinu medzi jej váhou a novou pamäťovou bunkou. Nakoniec vypočítame skrytý stav bunky ako súčin výstupnej brány a novej pamäťovej bunky, na ktorý je aplikovaná aktivačná funkcia ReLU alebo tanh. Výstup tejto ConvLSTMCell je nová pamäťová bunka a skrytý stav bunky. Na obrázku 6 je zobrazená zjednodušená štruktúra ConvLSTM bunky.



Obr. 6: Bunka ConvLSTM. [19]

Nasledujúce rovnice 1 detailne popisujú fungovanie ConvLSTM bunky, ako uvádzajú autori v článku [13].

$$\begin{aligned}
 i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f) \\
 C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o) \\
 H_t &= o_t \odot \tanh(C_t)
 \end{aligned} \tag{1}$$

Kde:

- $i_t$  je vstupná brána, ktorá kontroluje, akú novú informáciu máme pridať do pamäťovej bunky,
- $f_t$  je brána zabudnutia, ktorá kontroluje, akú časť bunky máme zabudnúť,
- $C_t$  je nový stav pamäťovej bunky,
- $o_t$  je výstupná brána, ktorá reguluje, akú časť aktuálnej bunky použijeme na výstup,
- $H_t$  je skrytý stav, ktorý reprezentuje výstup z bunky v čase  $t$ ,
- $W_{xg}, W_{xi}, W_{xf}, W_{xo}$  sú váhy pre vstupné dátá,
- $W_{hg}, W_{hi}, W_{hf}, W_{ho}$  sú váhy pre skryté stavy,
- $W_{ci}, W_{cf}, W_{co}$  sú váhy pre jednotlivé brány,
- $b_i, b_f, b_c, b_o$  sú biasy,
- $\sigma$  je aktivačná funkcia Sigmoid,
- $\tanh$  je hyperbolický tangens,
- $*$  predstavuje operáciu konvolúcie,
- $\odot$  predstavuje Hadamardov súčin.

**ConvLSTM:** Tento model je navrhnutý na spracovanie sekvencií obrázkov v čase. Je zložený z jednej vrstvy ConvLSTMCell, ktorá umožňuje učenie vzťahov medzi jednotlivými obrázkami v sekvencii. V každom kroku spracovania sa vstupná sekvencia obrázkov postupne odvíja cez časové kroky. Pre každý časový krok sa volá ConvLSTMCell, ktorá prijíma aktuálny obrázok zo sekvencie, predchádzajúci skrytý stav a stav pamäťovej bunky. Výstup z ConvLSTMCell je použitý ako skrytý stav a stav pamäťovej bunky pre ďalší časový krok a skryté stavy sú zaznamenané do výstupného tenzora.

**Seq2Seq:** Toto je hlavný model, ktorý využíva viacnásobné vrstvy ConvLSTM nasledované vrstvami batch normalizácie. Predikuje nasledujúci snímok v sekvencii na základe skrytých stavov generovaných vrstvami ConvLSTM. Prvá vrstva ConvLSTM má odlišné vstupné kanály, pretože táto vrstva prijíma priamo vstupné kanály obrázkov, zatiaľ čo ostatné vrstvy ConvLSTM v sekvencii prijímajú vstupné kanály zo skrytých stavov predchádzajúcich vrstiev. Posledná vrstva ConvLSTM je nasledovaná konvolučnou vrstvou, ktorá predikuje výstupný obrázok, a na zaistenie hodnôt pixelov medzi 0 a 1 sa použije aktivačná funkcia Sigmoid.

Zdrojový kód tohto modelu bol prevzatý z tutoriálu s názvom „Video Frame Prediction using ConvLSTM Network in PyTorch“ [9], ktorý sme upravili, aby bol schopný pracovať s našimi dátami.

### 3.1.2 Tréning

V rámci úpravy kódu sme implementovali vlastnú triedu pre dataset v knižnici PyTorch [10]. Pôvodný model bol navrhnutý pre MNIST dataset, ktorý sa skladal z 10 000 vzoriek sekvencií, z ktorých každá obsahuje 20 čiernobielych obrázkov veľkosti 64x64 pixelov. Namiesto viacerých odlišných sekvencií, ako v prípade MNIST datasetu, sme mali jednu kontinuálnu sekvenciu, kde sme mohli vybrať ľubovoľnú sériu po sebe nasledujúcich obrázkov z celého datasetu, pričom sekvencie sa mohli aj prekrývať. Preto sme upravili spôsob, akým sme získavali dátu z datasetu, aby sme mohli pracovať s touto novou štruktúrou.

Okrem toho sme implementovali transformačnú funkciu, ktorá upravovala naše dátá podľa popisu v sekcii 2.2 Spracovanie dát. Naše dátá boli farebné obrázky s troma farebnými kanálmi, na rozdiel od MNIST datasetu, ktorý obsahuje čiernobiele obrázky s jedným farebným kanálom. Preto sme museli upraviť aj spôsob, ako sme načítavalí dátá.

Experimentovali sme s rôznymi stratovými funkciami a ich kombináciami, vrátane Mean Squared Error (MSE) a Percepčnej stratovej funkciou [20]. Taktiež sme experimentovali s rôznym počtom predikovaných obrázkov na výstupe a s časovým rozostupom medzi vstupnými obrázkami.

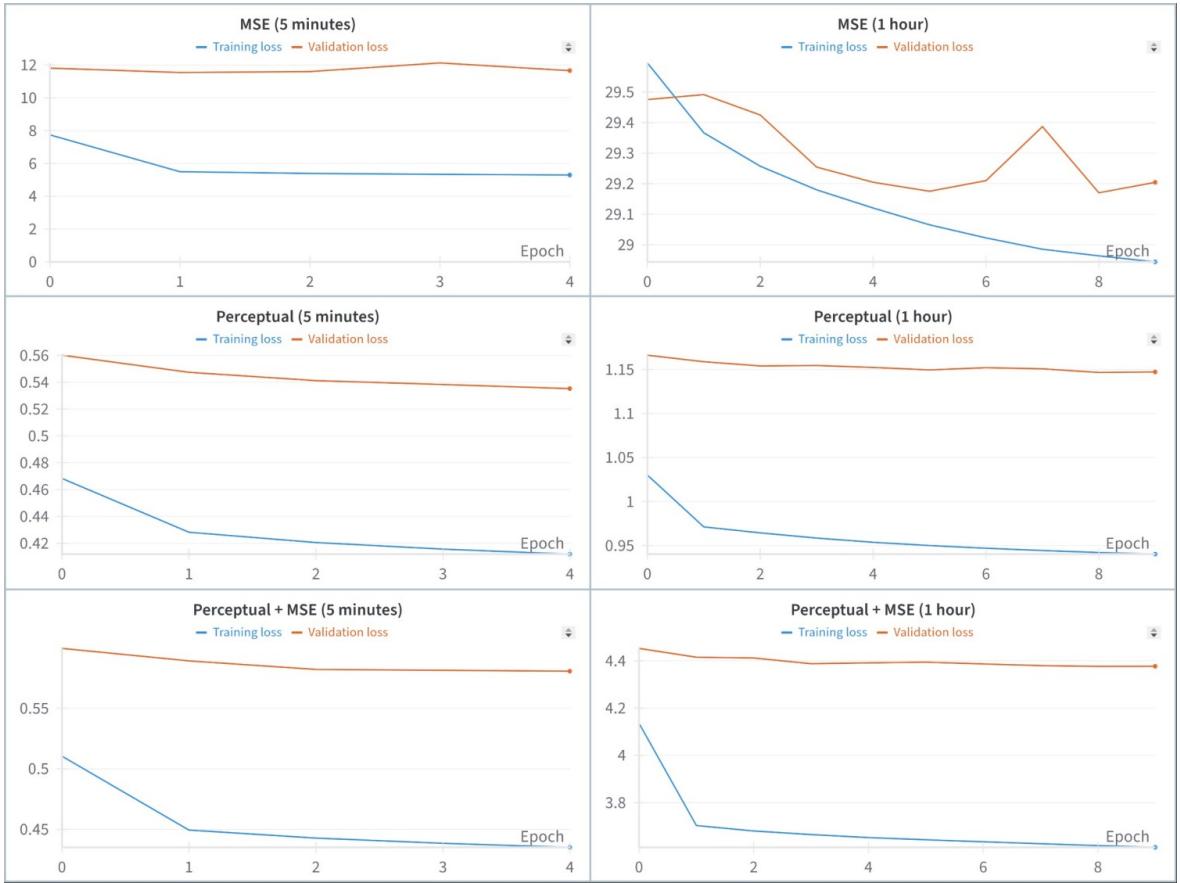
V dôsledku týchto experimentov sme natrénovali tri modely: jeden s použitím Mean Squared Error (MSE), druhý s použitím Percepčnej stratovej funkcie, a tretí model s kombináciou Percepčnej stratovej funkcie a MSE. Všetky tri modely boli trénované na sekvenciách s časovým rozostupom 5 minút, ako aj na sekvenciách s časovým rozostupom 1 hodina. Nakoniec sme tiež implementovali model, ktorý predikoval nie jeden, ale štyri nasledujúce obrázky.

Pre modely trénované s časovým rozostupom 5 minút sme využili celkovo 50 000 obrázkov. Z toho 40 000 obrázkov sme použili na trénovanie, ďalších 5 000 obrázkov na validáciu a 5 000 obrázkov sme vyhradili pre testovanie. Tieto dátá pokrývali obdobie od 24. septembra 2021, 00:00 do 16. marca 2022, 14:40. Skúsili sme tiež použiť 100 000 obrázkov namiesto pôvodných 50 000, avšak model dosiahol podobné výsledky ako pri menšom počte dát, čo naznačuje, že zvýšenie počtu trénovacích dát nemalo významný vplyv na výsledok. V prípade modelov s časovým rozostupom 1 hodina sme takisto pracovali s celkovým počtom 50 000 obrázkov. Z toho sme 40 000 obrázkov použili na trénovanie, ďalších 5 000 obrázkov sme použili na validáciu a rovnaký počet obrázkov

bol vyhradený pre testovanie. Tieto dátá pokrývali obdobie od 1. januára 2016, 00:00 do 14. septembra 2021, 8:00.

Všetky modely mali jednu vrstvu s ConvLSTM bunkami. Celkový počet ConvLSTM buniek bol rovnaký ako dĺžka vstupnej sekvencie obrázkov. Počet vstupných kanálov sme nastavili na 3, čo zodpovedá počtu farebných kanálov v obrázkoch a počet konvolučných jadier sme zvolili na 32 s rozmerom (3, 3). Využívali sme aktivačnú funkciu ReLU a vstupné obrázky mali rozmer 517x288 pixelov. Pre modely, ktoré predikovali iba jeden výstupný obrázok, sme zvolili batch size 8 a pre model, ktorý predikoval štyri nasledujúce obrázky, sme zvolili batch size 1. Skúšali sme rôzne konfigurácie, ale kvôli pamäťovým obmedzeniam sme sa rozhodli pre túto konkrétnu sadu parametrov.

Modely trénované na sekvenciách s časovým rozostupom 5 minút boli trénované po dobu 5 epoch, zatiaľ čo modely trénované na sekvenciách s časovým rozostupom 1 hodina boli trénované po dobu 10 epoch. Jedna epocha trvala približne 1 a pol hodiny pre modely s 5-minútovým rozostupom a 45 minút pre modely s hodinovým rozostupom, takže celkový tréning jedného modelu trval približne 7 a pol hodiny. Na začiatku trénovania bolo zaznamenané rýchle klesanie hodnôt stratovej funkcie, čo naznačuje efektívne učenie sa modelu. Každá ďalšia epocha naznačovala spomaľovanie procesu učenia a konvergenciu k určitej hodnote stratovej funkcie, tým pádom ďalšie učenie už nebolo tak efektívne. Na obrázku 7 sú zobrazené grafy vývoja trénovacej a validačnej straty, ktoré ilustrujú spomalenie poklesu straty a s každou ďalšou epochou naznačujú mierne preučenie. Napriek zdanlivo pozitívnym hodnotám straty, výsledky v nasledujúcej kapitole 4, nie sú práve optimálne a nezodpovedajú očakávaniam.

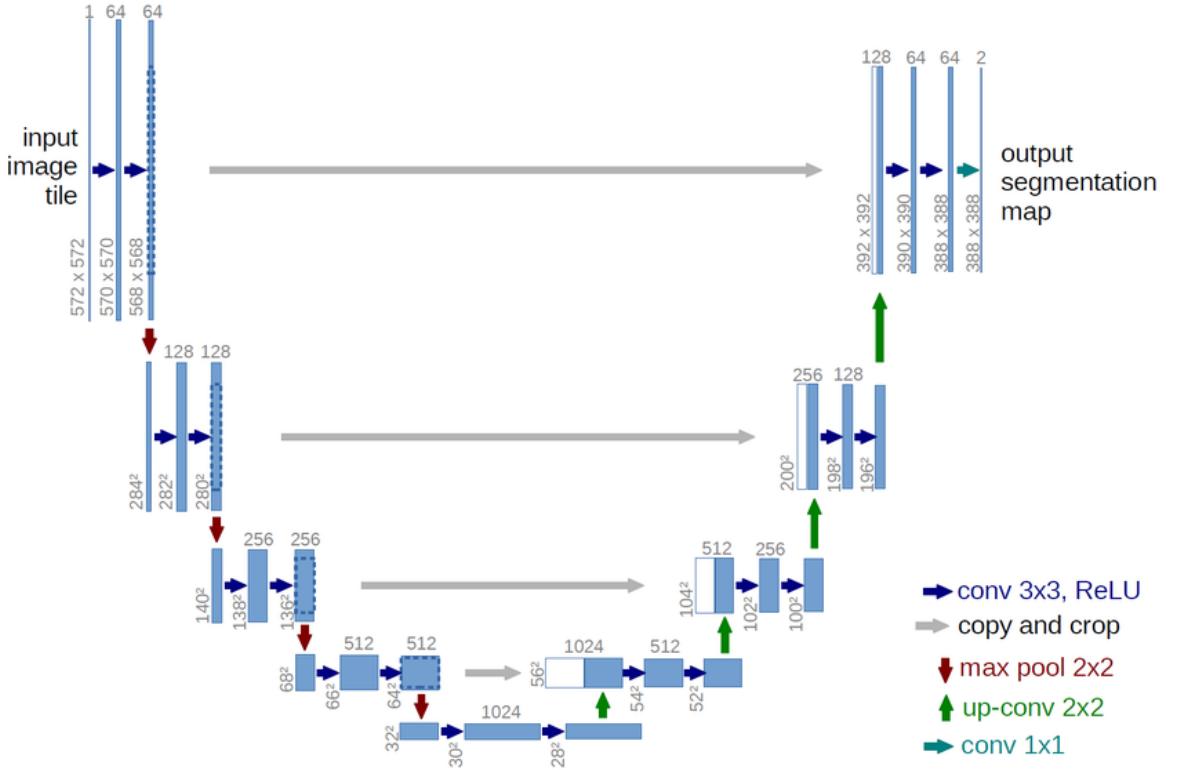


Obr. 7: Grafy trénovacích a validačných strát pre modely, ktoré predikovali jeden výstupný obrázok, s rôznymi stratovými funkciami (MSE, Percepčná stratová funkcia a ich kombinácia) trénované na sekvenciach s časovým rozostupom 5 minút, ako aj na sekvenciách s časovým rozostupom 1 hodina.

## 3.2 Rozšírenie ConvLSTM o enkóder a dekodér

### 3.2.1 Architektúra

U-Net je architektúra neurónovej siete, ktorá bola pôvodne navrhnutá na segmentáciu biomedicínskych obrázkov. Bola predstavená v článku „U-Net: Convolutional Networks for Biomedical Image Segmentation“ [11]. U-Net pozostáva z dvoch hlavných častí: enkódera a dekodéra. Enkóder slúži na extrakciu príznakov zo vstupného obrázka a redukciu jeho rozmerov, zatiaľ čo dekodér rekonštruuje obrázok pomocou informácií získaných z enkódera a zväčšuje rozmery obrázka na pôvodnú veľkosť. Jeho charakteristickým rysom sú tzv. „skip-connections“, ktoré umožňujú prenášať informácie z enkódera priamo do dekodéra, čím sa zabezpečuje presnejšia rekonštrukcia obrázkov. Ukážka architektúry U-Net je na obrázku 8.



Obr. 8: U-Net architektúra (príklad pre najnižšie rozlíšenie  $32 \times 32$  pixelov). Každý modrý box zodpovedá viackanálovej príznakovovej mape. Počet kanálov je uvedený na vrchu boxu. Veľkosti  $x, y$  sú uvedené v ľavom dolnom rohu boxu. Biele boxy predstavujú skopírované príznakové mapy. Šípky označujú rôzne operácie. [11]

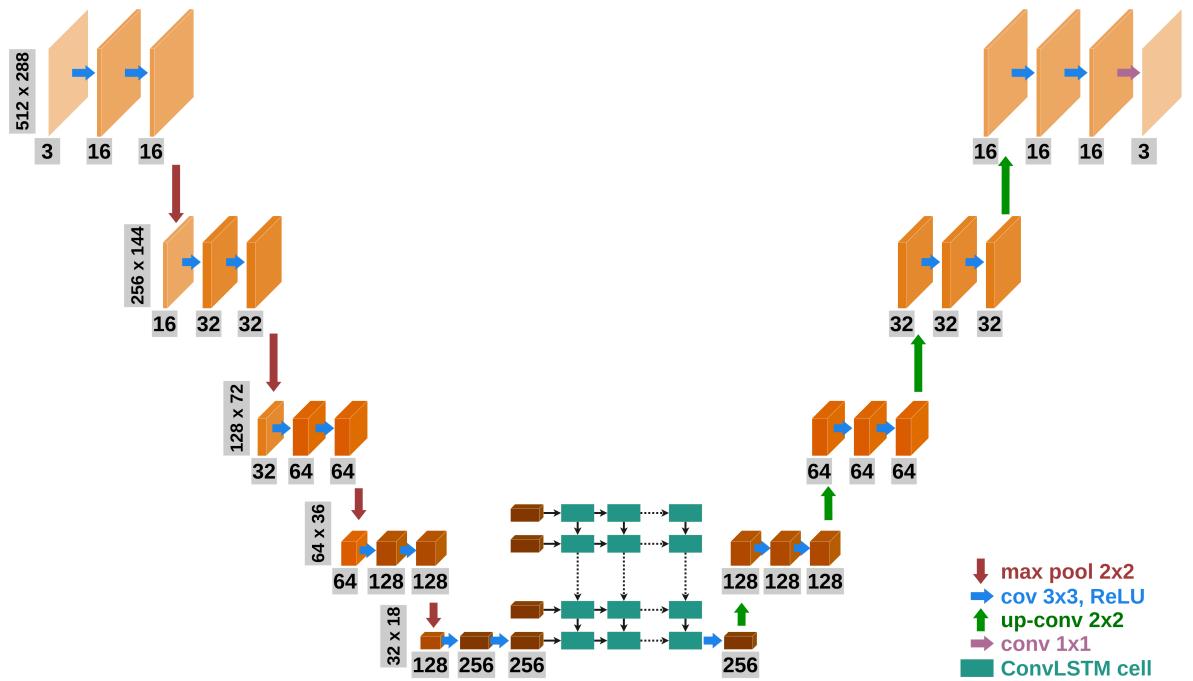
V našej implementácii modelu sme sa rozhodli použiť enkóder a dekadér odseparované. Enkóder pozostával z piatich blokov, pričom každý blok obsahoval dve konvolučné vrstvy nasledované max-poolingu vrstvou, s výnimkou posledného bloku, ktorý už max-poolingu vrstvu neobsahoval. Keďže každá max-pooling vrstva redukuje rozmer obrázka na polovicu, výstupy z enkódera boli 16-krát menšie ako vstupné obrázky a obsahovali 256 kanálov. Dekadér pozostával zo štyroch blokov, z ktorých každý obsahoval jednu upsampling vrstvu a dve konvolučné vrstvy. Upsampling vrstvy boli použité na postupné zväčšenie rozmerov obrázka na pôvodnú veľkosť. Na konci sa ešte použila konvolučná vrstva s jadrom veľkosti  $1 \times 1$  na získanie výsledného obrázka.

Inšpirovali sme sa kódom a postupom popísaným v článku zverejnenom na Towards Data Science [17], ktorý nám poskytol užitočné informácie a návod na implementáciu modelu U-Net s použitím knižnice PyTorch.

V našej implementácii sme spojili enkóder a dekadér s ConvLSTM architektúrou popísanou v sekcii 3.1 ConvLSTM, ako je vidieť na obrázku 9. Každý obrázok zo vstupnej sekvenčie 20 obrázkov bol v rámci for cyklu individuálne spracovaný enkóderom,

ktorý zmenšil ich rozmery. Tieto zmenšené obrázky (príznakové mapy) predstavovali vstupnú sekvenciu ConvLSTM architektúry, ktorá predikovala nasledujúci obrázok v sekvencii. Celkový počet ConvLSTM buniek zodpovedal počtu obrázkov vo vstupnej sekvencii krát počet vrstiev ConvLSTM. Výstup z ConvLSTM potom slúžil ako vstup do dekodéra, ktorý zrekonštruoval predikovaný obrázok na pôvodnú veľkosť. Pre jednoduchosť sme sa rozhodli vyniechať skip-connections.

Tento model sme navrhli s cieľom znížiť výpočtovú náročnosť a dosiahnuť efektívne spracovanie časovo závislých dát. Pôvodné vstupné dátá mali rozmer 20 x 3 x 512 x 288, čo predstavuje 20 obrázkov v sekvencii, každý s troma kanálmi a s rozmerom 512x288 pixelov. Celkový počet pixelov v pôvodných dátach bol teda 8 847 360. S využitím enkódera a dekodéra sa veľkosť dát pred vstupom do ConvLSTM zmení na 20 x 256 x 32 x 18, čo zodpovedá celkovému počtu 2 949 120 pixelov. Tento prístup nám umožnil redukovať veľkosť vstupných dát na tretinu a vďaka tomu sme boli schopní použiť viac ConvLSTM vrstiev.



Obr. 9: Model ConvLSTM spojený s enkóderom a dekodérom. Každý oranžový box zodpovedá viackanálovej príznakovovej mape. Počet kanálov je uvedený pod boxom. Rozmery sú uvedené na ľavej strane a korešpondujú s každým boxom v danej úrovni. Modrozelené boxy predstavujú ConvLSTM bunky. Šípky označujú rôzne operácie. Pre prehľadnosť je zobrazené enkódovanie iba jedného obrázka.

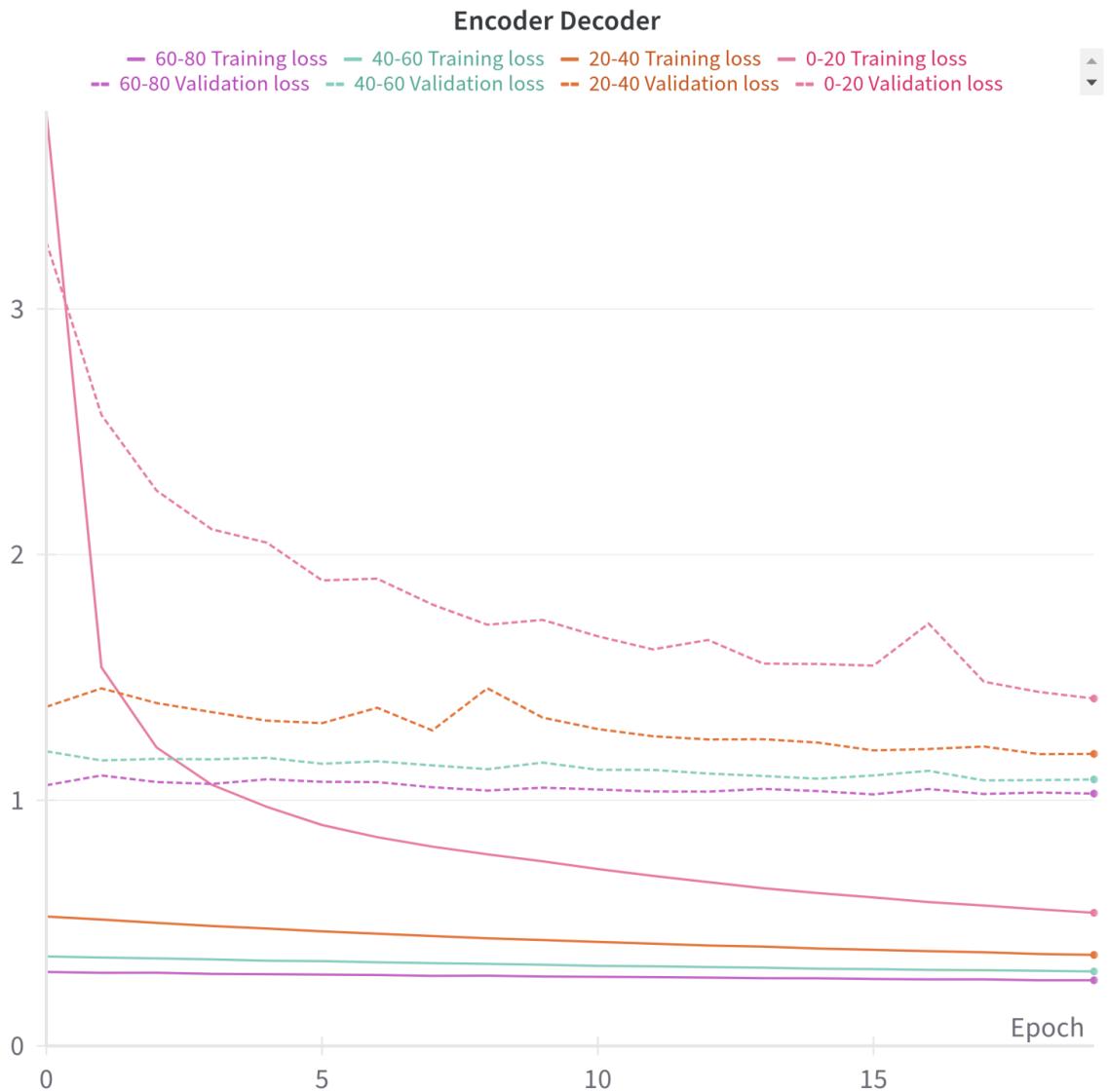
### 3.2.2 Tréning

Pred samotným tréningom sme upravili transformačnú funkciu v datasete tak, aby boli obrázky orezané na veľkosť 512x288 pixelov namesto pôvodných 517x288 pixelov. Tento krok bol nevyhnutný kvôli použitiu max-pooling vrstiev v architektúre, ktoré redukovali rozmery obrázkov na polovicu.

Použili sme celkovo 80 000 obrázkov, z toho 64 000 obrázkov sme vyhradili na trénovanie, ďalších 8 000 obrázkov na validáciu a rovnaký počet obrázkov sme použili na testovanie. Tieto dátu pokrývali časové obdobie od 24. septembra 2021, 00:00 do 18. júna 2022, 18:40 s časovým rozostupom 5 minút.

Najprv sme trénovali enkóder spolu s dekodérom ako jednotný model, ktorý dostal na vstup jeden obrázok a výstupom bol tiež len jeden obrázok. Cieľom bolo, aby výstupný obrázok bol čo najviac podobný tomu vstupnému.

Tréning prebiehal po dobu 80 epoch a rozdelili sme ho na štyri fázy, kde sme každú fazu spustili na 20 epochách. Celkový tréning tohto modelu trval približne 16 hodín, čo znamená približne 12 minút na každú epochu. Počas tréningu sme sledovali vývoj trénovacej a validačnej straty, ktorá postupne klesala, aj keď ku koncu tréningu už len veľmi pomaly. Trénovacie a validačné straty počas tréningu enkódera a dekodéra sú zobrazené na obrázku 10.

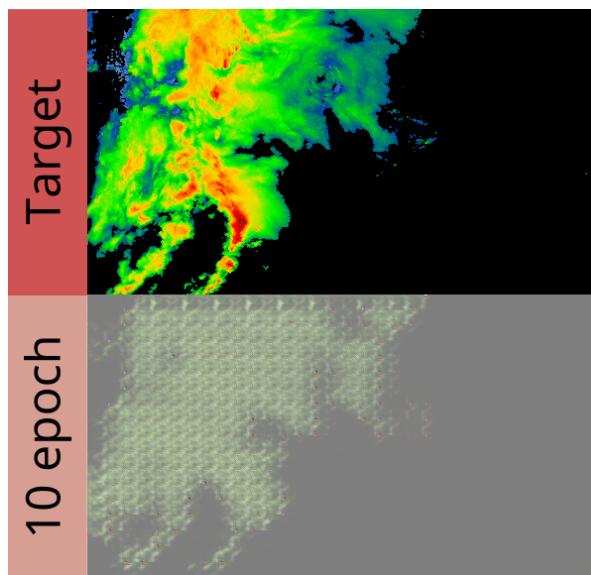


Obr. 10: Graf trénovacích a validačných strát pri tréningu enkódera a dekodéra.

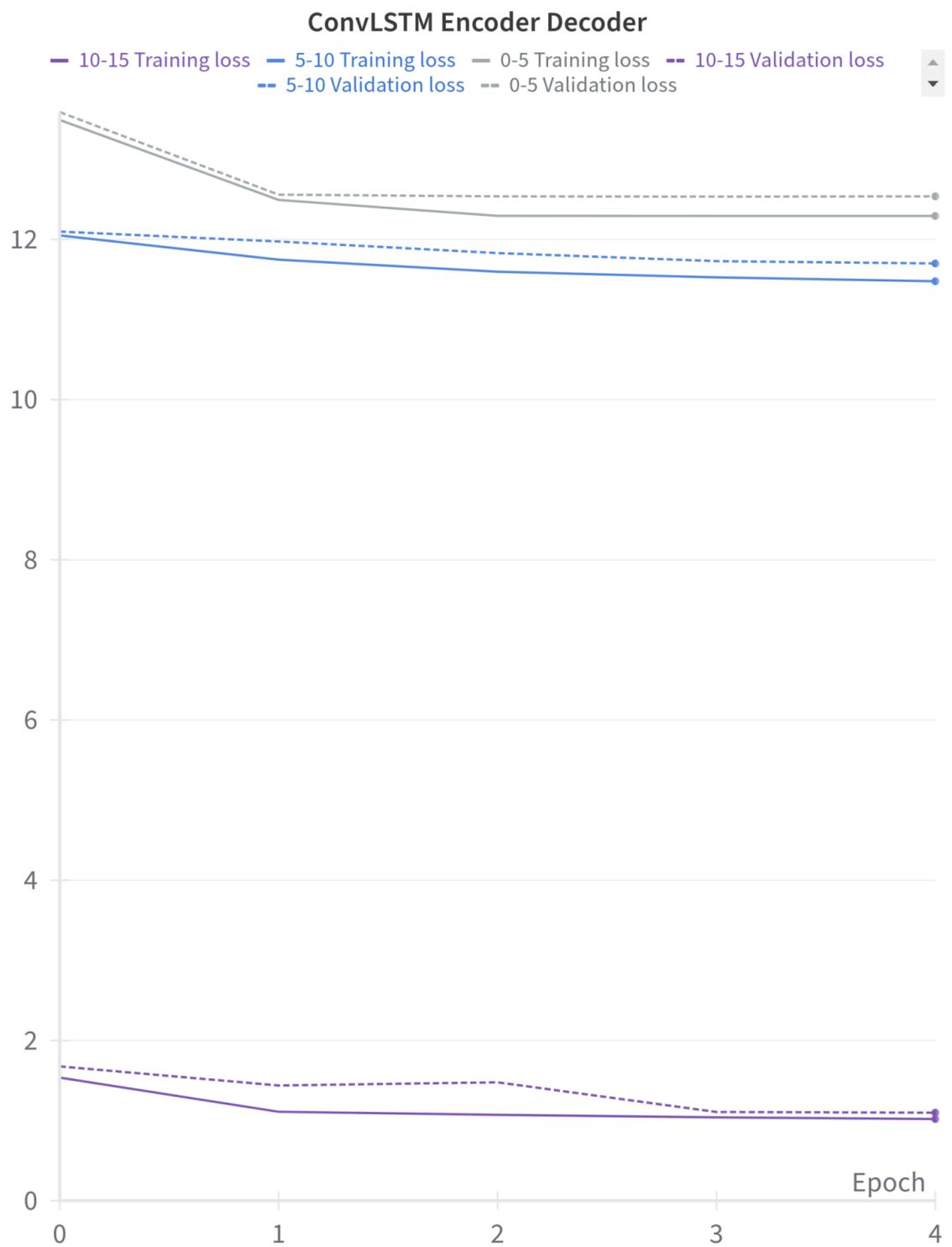
Po dokončení tohto tréningu sme využili vrstvy enkódera a dekodéra takto predtrénovaného modelu samostatne vo výslednom modeli, ktorý už zahŕňal aj ConvLSTM architektúru. Výsledný model obsahoval 8 vrstiev s ConvLSTM bunkami, pričom každá vrstva pozostávala z 20 takýchto buniek, čo zodpovedalo počtu obrázkov vo vstupnej sekvencii. Počet vstupných kanálov pre ConvLSTM sme nastavili na 256, čo sa zhodovalo s počtom výstupných kanálov po enkódovaní a počet konvolučných jadier sme zvolili na 128 s rozmerom (3, 3). Ako aktivačnú funkciu pre výstupy z ConvLSTM buniek sme použili ReLU a vstupné obrázky mali rozmer 512x288 pixelov. Batch size sme nastavili na 24.

Tréning sme rozdelili do 3 fáz, z ktorých každá trvala po dobu 5 epoch. Počas prvých dvoch fáz sme pozastavili aktualizáciu váh enkódera a dekodéra, aby sme

zabránili ich preučeniu. Keď stratové funkcie naznačovali, že model už nedosahuje ďalšie zlepšenia vo svojich výsledkoch, povolili sme aktualizáciu váh enkódera a de-kodéra a spustili poslednú fázu tréningu. Po prvých dvoch fázach model predikoval menej kvalitné obrázky, kde sa namiesto čierneho pozadia objavovala šedá farba a pôsobili transparentne, ako môžeme vidieť na obrázku 11. Po tretej fáze sa stratová funkcia výrazne zlepšila a model už predikoval obrázky s čiernym pozadím. Tento tréning trval približne 18 hodín, čo je v priemere 1 hodina a 12 minút na každú epochu. Trénovacie a validačné straty počas tohto tréningu sú zobrazené na obrázku 12.



Obr. 11: Výstup z modelu po prvých 10 epochách.



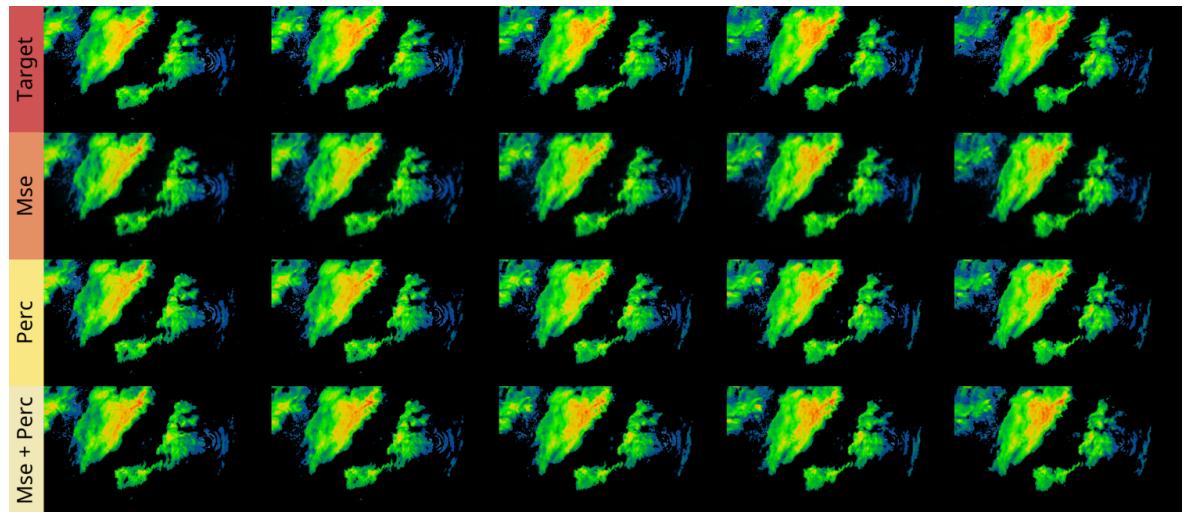
Obr. 12: Graf trénovacích a validačných strát pri tréningu ConvLSTM siete rozšírenej o enkóder a dekodér

# 4 Výsledky

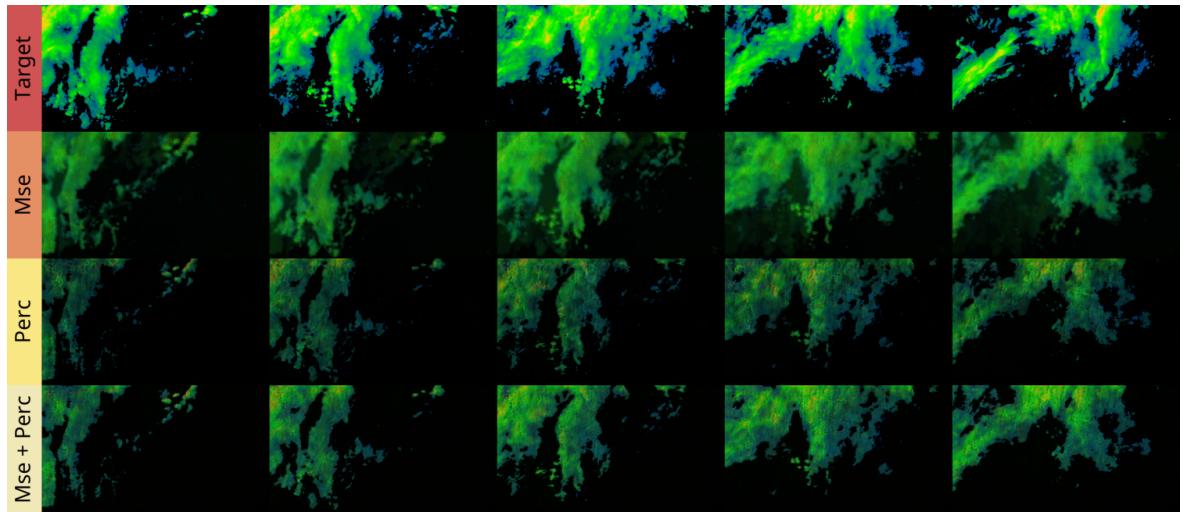
## 4.1 ConvLSTM

Každý z modelov sme trénovali a testovali so vstupnou sekvenčiou pozostávajúcou z 20 obrázkov. Počas našich experimentov sme zistili, že každý z modelov sa naučil predikovať posledný obrázok zo vstupnej sekvencie namiesto požadovaného cieľového obrázka. To znamená, že modely nedokázali úplne presne predikovať budúci stav, ale skôr opakovali posledný známy obrázok zo vstupnej sekvencie, čo je zrejmé z obrázkov 13 a 14. Každý obrázok z druhého, tretieho a štvrtého riadku sa podobal na cieľový obrázok z predchádzajúceho stĺpca, ktorý zodpovedá poslednému obrázku zo vstupnej sekvencie.

Čo sa týka stratových funkcií, MSE spôsobovalo rozmazanie predikovaného obrázka, zatiaľ čo Percepčná strata postupne znižovala intenzitu farieb. Kombinácia MSE a Percepčnej straty bola najlepšia pokial' ide o farby a ostrosť.

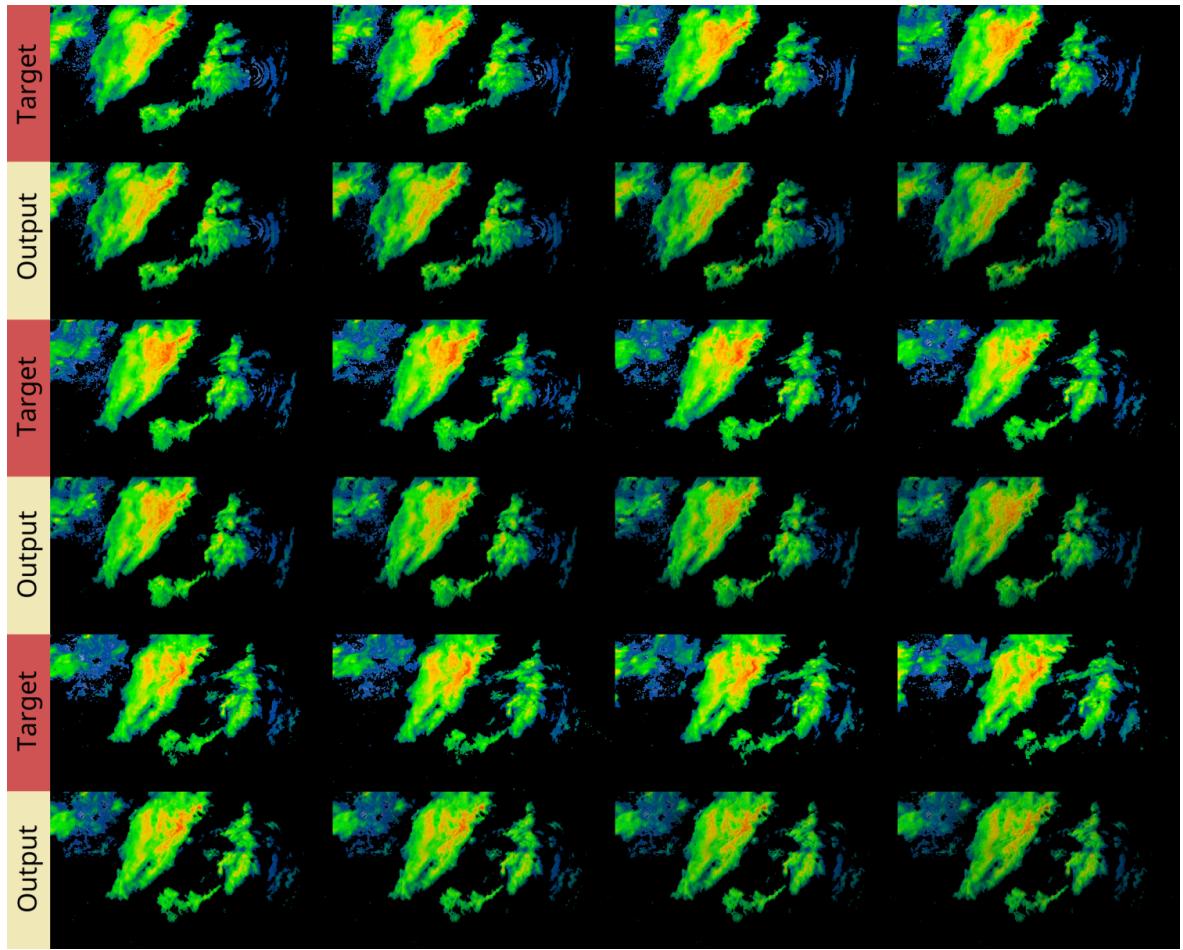


Obr. 13: Porovnanie medzi cieľovými obrázkami a predikciami modelu s použitím rôznych stratových funkcií (MSE, Percepčná stratová funkcia a kombinácia MSE a Percepčnej stratovej funkcie) pri časovom rozstupe 5 minút.



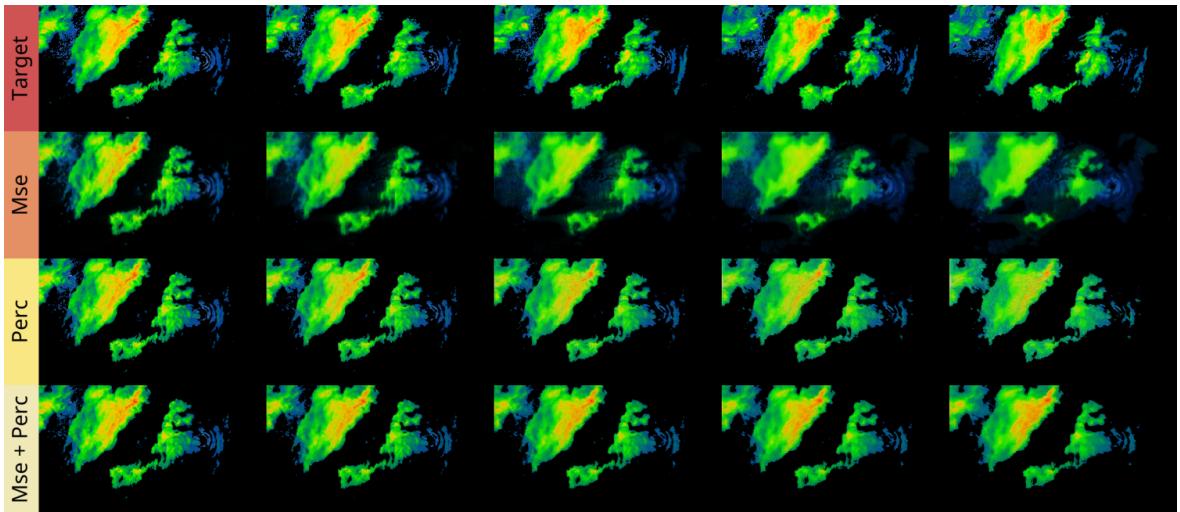
Obr. 14: Porovnanie medzi cieľovými obrázkami a predikciami modelu s použitím rôznych stratových funkcií (MSE, Percepčná stratová funkcia a kombinácia MSE a Percepčnej stratovej funkcie) pri časovom rozostupe 1 hodina.

Tento jav sa potvrdil aj pri modeli, ktorý predikoval 4 obrázky, čo je vidieť na obrázku 15, kde všetky 4 obrázky z výstupu mali rovnaký tvar, podobný poslednému cieľovému obrázku o tri riadky vyššie, ktorý zodpovedá poslednému obrázku zo vstupnej sekvencie. Tvary na týchto 4 predikovaných obrázkoch sa nehýbali, len postupne strácali na intenzite.

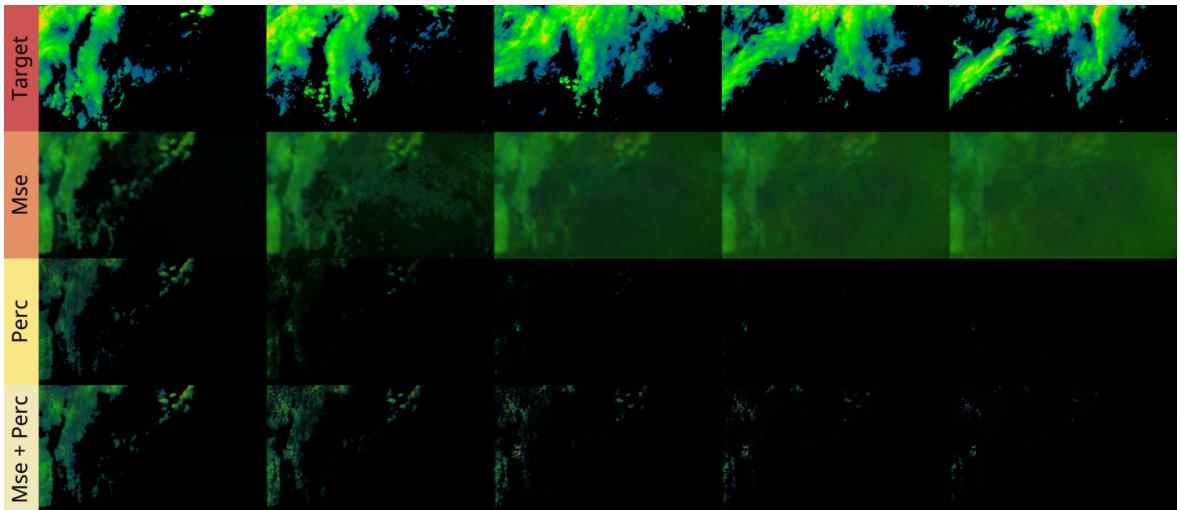


Obr. 15: Porovnanie medzi cielovými obrázkami a predikciami modelu, ktorý využíva kombináciu MSE a Percepčnej stratovej funkcie pri časovom rozostupe 5 minút a predikuje štyri obrázky naraz.

Na obrázkoch 16 a 17 sú zobrazené výstupy modelov, ktoré dostávali ako vstup predikcie z predchádzajúcich krokov namiesto originálnych obrázkov. Proces tvorby týchto výstupov spočíval v tom, že na začiatku sme predikovali zo vstupnej sekvenčie 20 originálnych obrázkov. Po každej predikcii sme odstránili prvý obrázok zo sekvenčie a výstup modelu sme vložili ako posledný obrázok. Túto novú sekvenciu sme následne vložili ako vstup pre ďalšiu predikciu modelu. Takto sme postupne aktualizovali vstupnú sekvenciu modelu s každou iteráciou predikcie. Je vidieť, že model s použitím MSE predikoval stále rozmazanejsie výstupy, zatiaľ čo model s Percepčnou stratou predikoval výstupy so slabšími farbami.



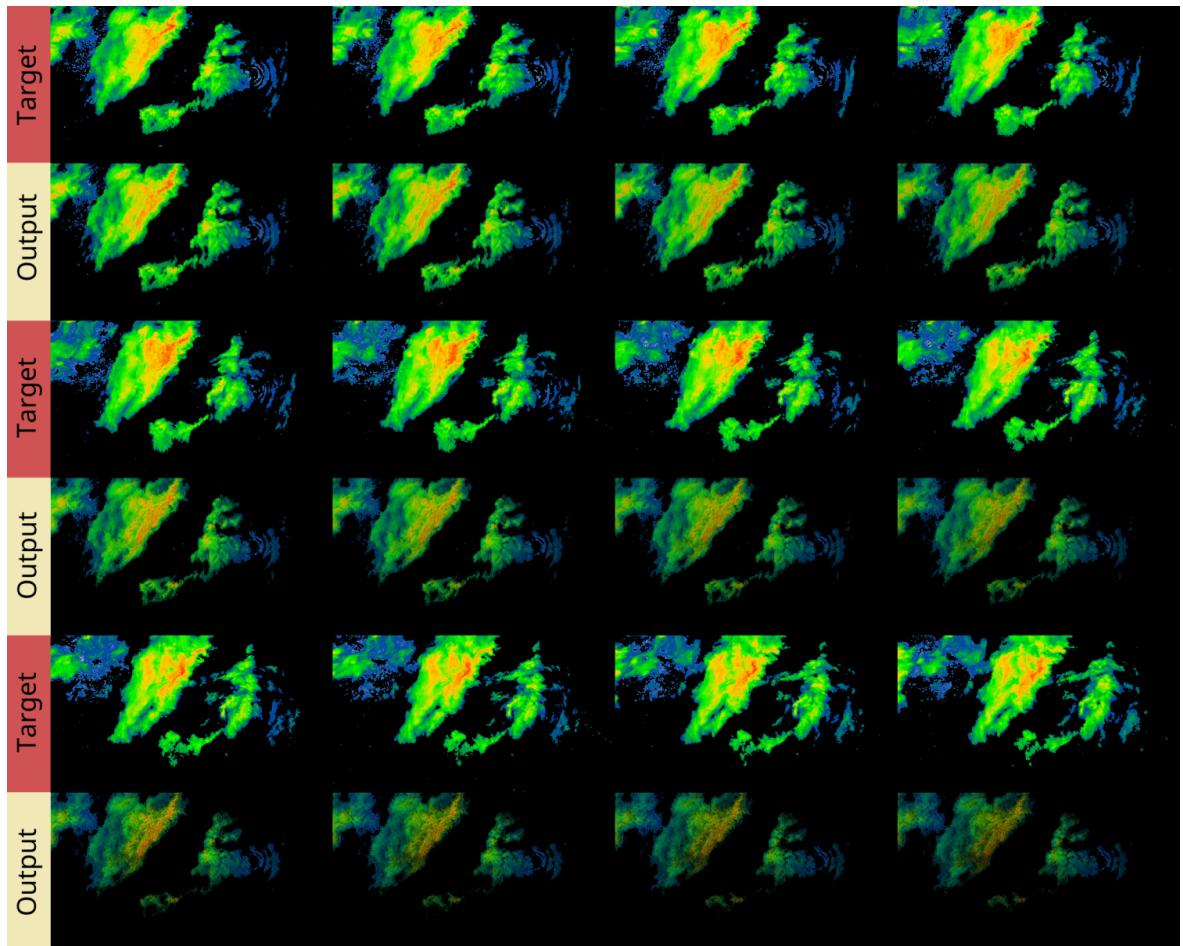
Obr. 16: Porovnanie medzi cieľovými obrázkami a predikciami modelu s použitím rôznych stratových funkcií (MSE, Percepčná stratová funkcia a kombinácia MSE a Percepčnej stratovej funkcie) pri časovom rozostupe 5 minút, pričom model dostáva ako vstup predikcie z predchádzajúcich krokov namiesto originálnych obrázkov.



Obr. 17: Porovnanie medzi cieľovými obrázkami a predikciami modelu s použitím rôznych stratových funkcií (MSE, Percepčná stratová funkcia a kombinácia MSE a Percepčnej stratovej funkcie) pri časovom rozostupe 1 hodina, pričom model dostáva ako vstup predikcie z predchádzajúcich krokov namiesto originálnych obrázkov.

Na obrázku 18 sú zobrazené výstupy modelu, ktorý predikoval 4 obrázky naraz a taktiež dostával ako vstup predikcie z predchádzajúcich krokov namiesto originálnych obrázkov. Postup tvorby týchto výstupov bol nasledovný: Začali sme predikciou zo vstupnej sekvencie 20 originálnych obrázkov. Po každej predikcii sme odstránili prvé 4

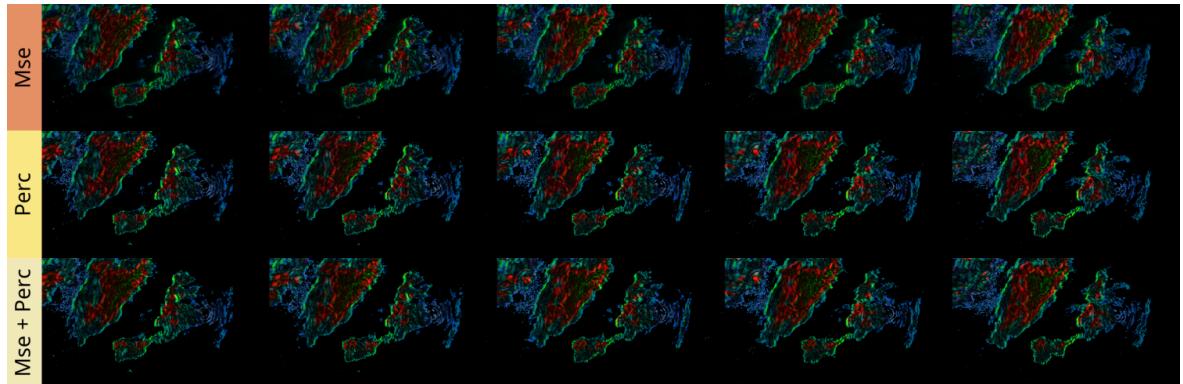
obrázky zo sekvencie a namiesto nich sme vložili výstupy modelu. Túto novú sekvenčiu sme potom využili ako vstup pre ďalšiu predikciu modelu. Celkovo sú na obrázku 18 zobrazené 3 sady 4-obrázkových výstupov siete spolu s odpovedajúcimi cieľovými obrázkami. Je na ňom vidieť, že predikovaný tvar sa nemenil, len postupne strácal na intenzite.



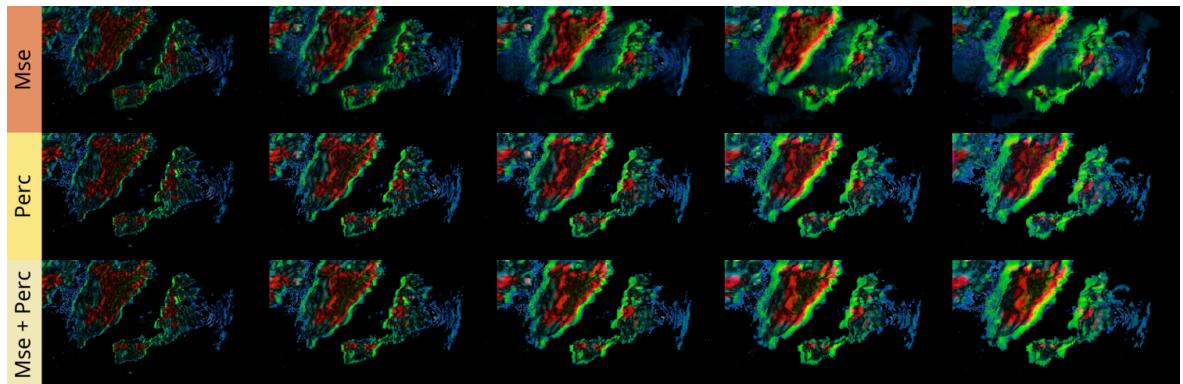
Obr. 18: Porovnanie medzi cieľovými obrázkami a predikciami modelu, ktorý využíva kombináciu MSE a Percepčnej stratovej funkcie pri časovom rozostupe 5 minút a predikuje štyri obrázky naraz, pričom vstupom sú predikcie z predchádzajúcich krokov namiesto originálnych obrázkov.

Na obrázku 19 sú znázornené rozdiely medzi predikovanými a cieľovými obrázkami, kde model dostával ako vstup originálne obrázky a na obrázku 20 sú znázornené rozdiely medzi predikovanými a cieľovými obrázkami, kde model dostával ako vstup predikcie z predchádzajúcich krokov. Tieto rozdiely sú získané pomocou operácie absolútnej hodnoty rozdielu pixelov medzi cieľovými obrázkami a jednotlivými výstupmi modelov a poskytujú nám informáciu o tom, ako sa výstupy modelov líšia od požadovaných cieľových obrázkov. V prípade, že by bol celý obrázok čierny, rozdiel by bol minimálny

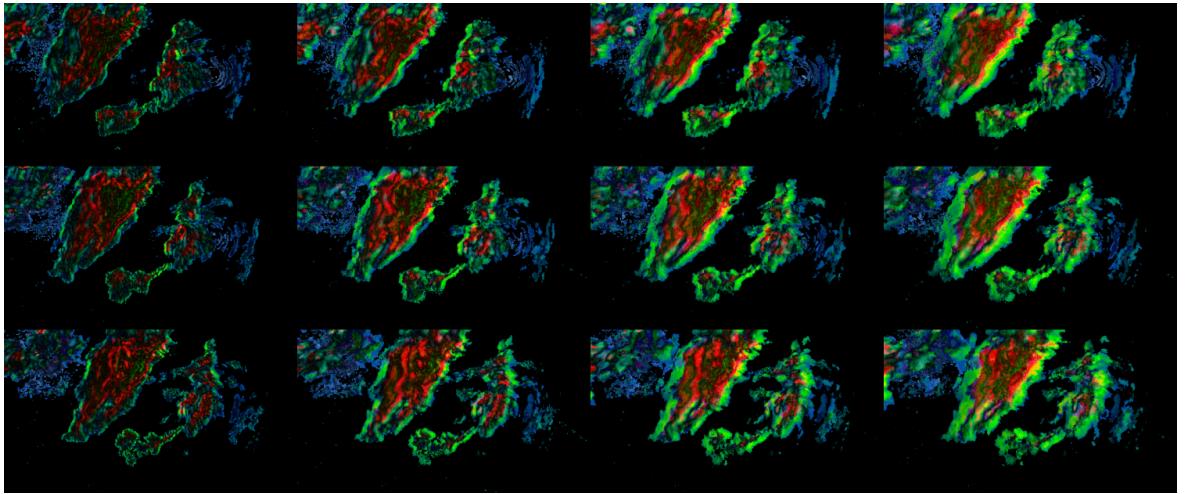
a model by dosiahol optimálne výsledky. Avšak čím viac farebných oblastí obsahujú, tým väčšie odchýlky existujú medzi predikciou a skutočnosťou, čo naznačuje nižšiu úspešnosť modelu. Z rozdielov na obrázku 20 môžeme usúdiť, že predikovaný obrázok sa nehýbal, zatiaľ čo cieľový obrázok sa postupne posúval doprava. Celkovo modely stále predikovali posledný obrázok zo vstupnej sekvencie, čo viedlo k opakovaniu toho istého tvaru. Túto skutočnosť je možné pozorovať aj na obrázku 21, kde model predikoval 4 obrázky naraz.



Obr. 19: Rozdiely medzi cieľovými obrázkami a predikciami modelu s použitím rôznych stratových funkcií (MSE, Percepčná stratová funkcia a kombinácia MSE a Percepčnej stratovej funkcie) pri časovom rozostupe 5 minút.



Obr. 20: Rozdiely medzi cieľovými obrázkami a predikciami modelu s použitím rôznych stratových funkcií (MSE, Percepčná stratová funkcia a kombinácia MSE a Percepčnej stratovej funkcie) pri časovom rozostupe 5 minút, pričom model dostáva ako vstup predikcie z predchádzajúcich krokov namiesto originálnych obrázkov.



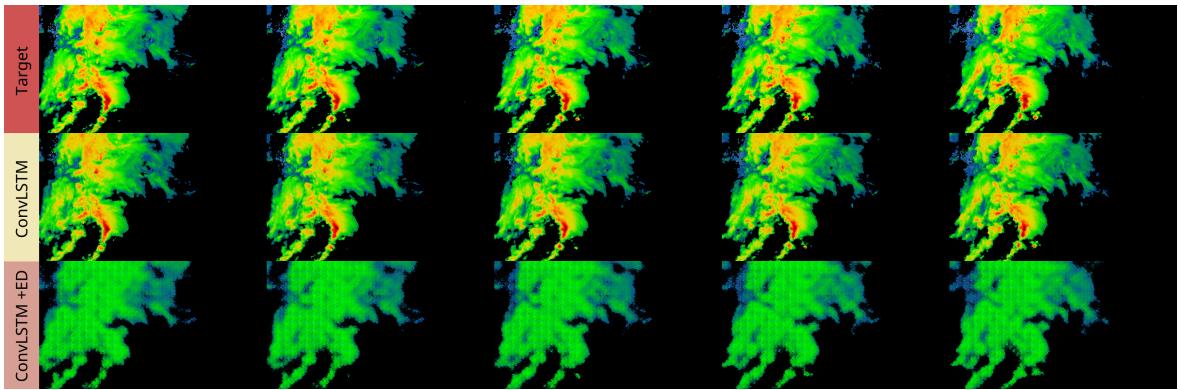
Obr. 21: Rozdiely medzi cieľovými obrázkami a predikciami modelu, ktorý využíva kombináciu MSE a Percepčnej stratovej funkcie pri časovom rozostupe 5 minút a predikuje štyri obrázky naraz.

## 4.2 Rozšírenie ConvLSTM o enkóder a dekodér

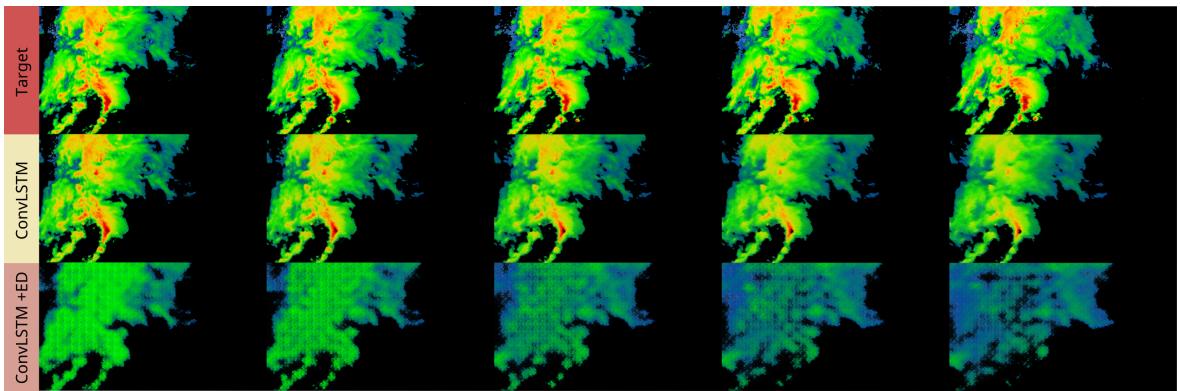
V tejto sekcií budeme analyzovať výsledky modelu rozšíreného o enkóder a dekodér, ktorý sme trénovali a testovali so vstupnou sekvenciou 20 obrázkov s časovým rozostupom 5 minút. Pri trénovaní tohto modelu sme využívali kombináciu MSE a Percepčnej straty. Tieto stratové funkcie sme zvolili na základe ich schopnosti minimalizovať rozmazenie predikovaných obrázkov a zachovať ostrosť detailov, ktorá sa potvrdila pri experimentoch v sekcií 4.1 ConvLSTM.

Výsledky tohto modelu budeme porovnávať s jedným z predchádzajúcich modelov, konkrétnie s modelom, ktorý bol tiež trénovaný na dátach s časovým rozostupom 5 minút a využíval rovnakú kombináciu stratových funkcií. Cieľom tejto porovnávacej analýzy je zhodnotiť, či rozšírenie ConvLSTM o enkóder a dekodér prináša zlepšenie vo výsledkoch predikcie oproti pôvodnému modelu.

Model s enkóderom a dekodérom predikoval obrázky bez žltej a červenej farby a ich kvalita bola zreteľne nižšia v porovnaní s predchádzajúcim modelom, ako možno vidieť na obrázku 22, ktorý zobrazuje výstupy oboch modelov. Na obrázku 23 sú zobrazené predikcie oboch modelov, kde ako vstup dostávali predikcie z predchádzajúcich krokov namiesto originálnych obrázkov. Proces tvorby týchto výstupov bol bližšie popísaný v sekcií 4.1 ConvLSTM. Vidíme, že pôvodný model stále opakoval rovnaký tvar, zatiaľ čo výstupy modelu s enkóderom a dekodérom sa v každom kroku líšili. Aj napriek tejto variabilite, boli výstupy stále menej kvalitné po každom kroku.

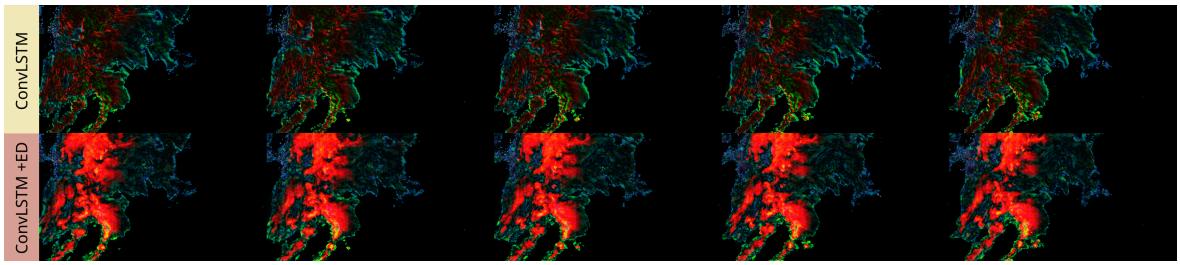


Obr. 22: Porovnanie medzi cielovými obrázkami a predikciami dvoch modelov, kde prvý model je ConvLSTM bez enkódera a dekodéra, zatiaľ čo druhý model je ConvLSTM s enkóderom a dekodérom.



Obr. 23: Porovnanie medzi cielovými obrázkami a predikciami dvoch modelov, kde prvý model je ConvLSTM bez enkódera a dekodéra, zatiaľ čo druhý model je ConvLSTM s enkóderom a dekodérom. Modely dostávali ako vstup predikcie z predchádzajúcich krokov namiesto originálnych obrázkov.

Na obrázku 24 sú zobrazené rozdiely medzi cielovými obrázkami a predikciami oboch porovnávaných modelov. Proces tvorby týchto rozdielov je taktiež popísaný v sekcií 4.1 ConvLSTM. Z týchto rozdielov možno posúdiť, že model s enkóderom a dekodérom vykazoval väčšie rozdiely, čo sa týka farby a intenzity, avšak lepšie zachovával tvar objektov. Tento fakt možno pozorovať na okrajoch objektov, ktoré sú na predikciách modelu s enkóderom a dekodérom v porovnaní s predikciami pôvodného modelu tenšie a menej výrazne.



Obr. 24: Rozdiely medzi cieľovými obrázkami a predikciami dvoch modelov, kde prvý model je ConvLSTM bez enkódera a dekodéra, zatiaľ čo druhý model je ConvLSTM s enkóderom a dekodérom.

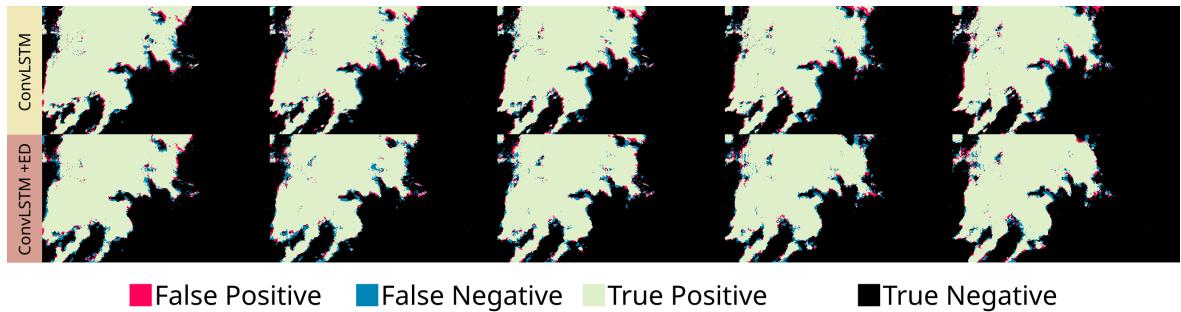
Pre ďalšiu analýzu sme sa rozhodli porovnať samotný tvar týchto objektov. Na tento účel sme previedli cieľové ako aj predikované obrázky na binárne, čo znamená, že sme ich pixelové hodnoty normalizovali do rozsahu 0-1, následne sme aplikovali binárny prah s hodnotou 0.4 a tento prah sme použili na konverziu pixelov na binárne hodnoty, kde 1 reprezentuje prítomnosť oblačnosti a 0 jej absenciu. Potom sme porovnali cieľové obrázky s predikovanými a vytvorili nové obrázky nastavením pixelov nasledovne:

- true negative - ak bol pixel na cieľovom obrázku aj na predikovanom obrázku 0, označili sme ho čierrou, čo znamená, že model korektne predikoval absenciú oblačnosti na danom mieste,
- false positive - ak bol pixel na cieľovom obrázku 0 a na predikcii 1, tento pixel sme označili červenou farbou, čím sme naznačili, že model predikoval oblačnosť na mieste, kde v skutočnosti nebola,
- false negative - ak pixel na cieľovom obrázku bol 1 a na predikcii bol 0, označili sme tento pixel modrou, čím sme naznačili, že model predikoval absenciú oblačnosti na mieste, kde v skutočnosti bola,
- true positive - ak bol pixel na oboch obrázkoch 1 (teda pixel reprezentoval oblačnosť), nastavili sme ho na svetlo žltú, čo znamená, že model korektne predikoval prítomnosť oblačnosti na danom mieste.

Tento spôsob porovnávania bol inšpirovaný článkom [1].

Na obrázku 25 je zobrazené porovnanie medzi cieľovými a predikovanými binárnymi obrázkami a môžeme pozorovať, že tvar oblačnosti predikovaný modelom s enkóderom a dekodérom má na okrajoch menej modrej aj červenej farby. Tento fakt naznačuje, že aj keď sa prvotne mohlo zdať, že tvar sa v dôsledku straty na intenzite a kvalite len zmenšoval, v skutočnosti sa pohyboval správnym smerom. Ak by sa len

zmenšoval a nepohyboval, sice by mal menej červenej na ľavom okraji, ale naopak by mal viac modrej na pravom. Z toho môžeme usúdiť, že model rozšírený o enkóder a dekódér vykazoval lepšie výsledky, pokiaľ ide o tvar a pohyb oblačnosti. Vďaka zníženiu výpočtovej náročnosti, ktoré zabezpečili enkóder a dekodér, sme mohli namiesto jednej ConvLSTM vrstvy s 32 konvolučnými jadrami použiť až 8 vrstiev s 128 konvolučnými jadrami. Tento prístup poskytol modelu väčší priestor na učenie, čo viedlo k lepšej schopnosti zachytiť a predikovať komplexné vzory v pohybe oblačnosti.



Obr. 25: Rozdiely medzi cieľovými a predikovanými binárnymi obrázkami dvoch modelov, kde prvý model je ConvLSTM bez enkódera a dekodéra, zatiaľ čo druhý model je ConvLSTM s enkóderom a dekodérom. Červená farba označuje miesta, kde model nesprávne predikoval prítomnosť oblačnosti, modrá farba označuje miesta, kde nesprávne predikoval absenciu oblačnosti, čierna farba označuje správne predikovanú absenciu oblačnosti a svetložltá farba označuje správne predikovanú prítomnosť oblačnosti.

## Porovnanie modelov pomocou metrík

Pre porovnanie výsledkov týchto dvoch modelov sme použili metriky Structural Similarity Index Measure (SSIM) a Mean Absolute Error (MAE). MAE sa vypočíta ako priemer absolútnych hodnôt rozdielov medzi pixelmi cieľového a predikovaného obrázka. Hodnoty MAE sú vždy nezáporné, pričom nižšie hodnoty indikujú menšiu odchýlku a tým lepší výkon modelu. Jeho hodnota sa môže pohybovať od 0 (žiadna chyba) po maximálnu možnú hodnotu v závislosti od rozsahu hodnôt pixelov v obrázku. V našom prípade sme porovnávali normalizované obrázky, čo znamená, že maximálna hodnota MAE bola 1.

Na rozdiel od MAE, SSIM hodnotí kvalitu a podobnosť medzi dvoma obrázkami, pričom zohľadňuje ľudské vizuálne vnímanie. Porovnáva obrázky na základe troch kľúčových vlastností: jasu, kontrastu a štruktúry. Jas sa meria ako priemer hodnôt

všetkých pixelov

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i, \quad (2)$$

kontrast porovnáva štandardnú odchýlku (odmocninu z rozptylu) všetkých hodnôt pixelov

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2} \quad (3)$$

a pri porovnávaní štruktúry sa využíva zjednotený vzorec, ktorý delí vstupný signál jeho štandardnou odchýlkou

$$\frac{x - \mu_x}{\sigma_x}. \quad (4)$$

Porovnávacie funkcie sú vypočítané nasledovne:

$$\begin{aligned} l(x, y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \\ c(x, y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \\ s(x, y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \end{aligned} \quad (5)$$

kde  $l(x, y)$  predstavuje porovnávaciu funkciu jasu,  $c(x, y)$  porovnávaciu funkciu kontrastu a  $s(x, y)$  porovnávaciu funkciu štruktúry medzi dvomi obrázkami.  $C_i$  sú konštanty, ktoré zabezpečujú stabilitu v prípade, že menovateľ vzorca dosiahne hodnotu 0.  $x, y$  predstavujú porovnávane obrázky a  $\sigma_{xy}$  je definovaná ako

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y). \quad (6)$$

Výsledne SSIM skóre sa vypočíta nasledovne:

$$\text{SSIM} = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma, \quad (7)$$

kde  $\alpha > 0, \beta > 0, \gamma > 0$  určujú dôležitosť každej z metrík. [7]

Výsledky týchto metrík pre obidva modely sú uvedené v tabuľke 1. Hoci sa z pohľadu hodnôt metrík pri porovnávaní týchto modelov môže zdať, že model bez enkodéra a dekodéra dosahoval lepšie výsledky, v skutočnosti jeho výkon neboli optimálny, pretože sa naučil iba predpovedať posledný obrázok zo vstupnej sekvencie, čo v kontexte predikcie vývoja a distribúcie oblačnosti v čase a priestore nemalo veľkú hodnotu. Naopak, model s enkodérom a dekodérom dokázal zahrnúť aj predikciu pohybu oblačnosti, hoci trpel nedostatkami najmä čo sa týka kvality obrázkov.

Model	MAE	SSIM	MAE binary	SSIM binary
ConvLSTM	0.0183	0.8526	0.0314	0.8433
ConvLSTM s enkóderom a dekodérom	0.02822	0.8040	0.0342	0.8382

Tabuľka 1: Porovnanie modelov pomocou metrík. MAE binary a SSIM binary predstavujú porovnanie predikovaných a cieľových obrázkov v ich binárnej podobe, ako bolo ilustrované na obrázku 25.

# Záver

Predikcia oblačnosti hraje kľúčovú úlohu v našom každodennom živote a má zásadný vplyv na rôzne oblasti, ako sú napríklad poľnohospodárstvo, doprava a energetika. Tradičné metódy, ako sú satelitné a radarové techniky, či numerické modely, majú svoje obmedzenia, najmä pri krátkodobej predikcii. S nástupom neurónových sietí sa otvárajú nové možnosti v oblasti meteorológie vďaka ich schopnosti efektívne zachytávať a modelovať komplexné vzory v časových radoch, ako aj ich využitiu v oblasti analýzy a spracovania obrazu.

Cieľom tejto práce bolo preskúmať a analyzovať existujúce metódy predikcie oblačnosti a následne navrhnúť a implementovať vlastný model neurónovej siete pre túto úlohu. V analýze sme sa zamerali na tradičné metódy, ako aj na využitie neurónových sietí v tejto oblasti. V dostupnej literatúre sme sa stretli s architektúrou U-Net, ktorá vynikajúco zvláda prácu s veľkými obrazovými dátami, čo je kľúčová vlastnosť vzhľadom na charakter tejto problematiky. Následne sme navrhli vlastný model, ktorý využíval architektúru konvolučnej LSTM siete. Kvôli pamäťovým obmedzeniam sme mohli použiť len jednu vrstvu ConvLSTM, čo negatívne ovplyvnilo schopnosť modelu zachytiť komplexné časovo-priestorové vzory, a preto sa náš model naučil predikovať posledný obrázok zo vstupnej sekvencie namiesto cieľového obrázka. Experimentovali sme s rôznymi stratovými funkciemi a ich kombináciami, ako aj s rôznymi časovými rozostupmi medzi obrázkami v datasete (5 minút a 1 hodina). Okrem toho sme natrénovali aj model, ktorý predikoval 4 obrázky namiesto jedného. Aj tento model sa však naučil predikovať posledný obrázok zo vstupnej sekvencie (4-krát za sebou) namiesto požadovaného cieľového obrázka. Na základe týchto zistení sme rozšírili architektúru o enkóder a dekodér, pričom sme sa inšpirovali architektúrou U-Net. Tento nový model výrazne znížil výpočtovú náročnosť a umožnil nám využiť až 8 vrstiev ConvLSTM. Aj keď metriky nasvedčovali tomu, že pôvodný model dosahoval lepšie výsledky, vizuálne porovnanie výstupov modelov naznačovalo, že model s enkóderom a dekodérom lepšie predpovedal polohu a tvar oblačnosti. Naopak, kvalita obrázkov v porovnaní s predchádzajúcimi verziami modelu sa zhoršila. Tento rozdiel v kvalite by mohol byť spôsobený nepresnosťami v enkóderi a dekodéri.

Mitigácia týchto nepresností môže byť predmetom ďalšieho skúmania. Optimálizácia modelov na viackrokovú predikciu by mohla zlepšiť ich schopnosť predikovať vývoj oblačnosti v čase a priestore. Okrem toho by bolo zaujímavé vyskúšať nové typy neurónových sietí, ako je napríklad model Stable Video Diffusion, ktorý je navrhnutý pre generovanie videí z jedného statického obrázka [3]. Ďalším vylepšením pre budúci výskum by mohlo byť pridanie ďalších vstupných dát, ako sú napríklad informácie o smere vetra, teplote, vlhkosti, tlaku atď.

# Zoznam použitej literatúry

- [1] BERTHOMIER, L., PRADEL, B., AND PEREZ, L. Cloud cover nowcasting with deep learning. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)* (2020), IEEE, pp. 1–6.
- [2] BIEWALD, L. Experiment tracking with weights and biases, 2020. Software available from [wandb.com](https://wandb.com).
- [3] BLATTMANN, A., DOCKHORN, T., KULAL, S., MENDELEVITCH, D., KILIAN, M., LORENZ, D., LEVI, Y., ENGLISH, Z., VOLETI, V., LETTS, A., ET AL. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).
- [4] BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).
- [5] CLARK, A. Pillow (pil fork) documentation. Dostupné na internete: <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>, 2015.
- [6] CONTRIBUTORS, W. Numerical weather prediction — Wikipedia, the free encyclopedia. Dostupné na internete: [https://en.wikipedia.org/w/index.php?title=Numerical\\_weather\\_prediction&oldid=1185267885](https://en.wikipedia.org/w/index.php?title=Numerical_weather_prediction&oldid=1185267885), 2023. [cit. 15. 5. 2024].
- [7] DATTA, P. All about structural similarity index (ssim): Theory + code in pytorch. Dostupné na internete: <https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e>, 2020. [cit. 15. 5. 2024].
- [8] OF METEROROLOGY, A. B. How does a weather radar work? Dostupné na internete: <https://media.bom.gov.au/social/blog/1459/how-does-a-weather-radar-work/>, 2017. [cit. 15. 5. 2024].
- [9] PANDA, R. Video frame prediction using convlstm network in pytorch. Dostupné na internete: <https://sladewinter.medium.com/video-frame-prediction-using-convlstm-network-in-pytorch-b5210a6ce582/>, 2021. [cit. 15. 5. 2024].

- [10] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., ET AL. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [11] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18 (2015), Springer, pp. 234–241.
- [12] SHI, E., LI, Q., GU, D., AND ZHAO, Z. Convolutional neural networks applied on weather radar echo extrapolation. *DESTech Trans. Comput. Sci. Eng* (2017).
- [13] SHI, X., CHEN, Z., WANG, H., YEUNG, D.-Y., WONG, W.-K., AND WOO, W.-C. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).
- [14] SHMÚ. Slovenská rádiolokačná siet. Dostupné na internete: <https://www.shmu.sk/sk/?page=1566>, 2024. [cit. 15. 5. 2024].
- [15] SINGH, K. Types of satellite imagery. Dostupné na internete: <https://pangeography.com/types-of-satellite-imagery/>, 2023. [cit. 15. 5. 2024].
- [16] TUYEN, D. N., TUAN, T. M., LE, X.-H., TUNG, N. T., CHAU, T. K., VAN HAI, P., GEROGIANNIS, V. C., AND SON, L. H. Rainpredrnn: A new approach for precipitation nowcasting with weather radar echo images based on deep learning. *Axioms* 11, 3 (2022), 107.
- [17] WAEL, M. Cook your first u-net in pytorch. Dostupné na internete: <https://towardsdatascience.com/cook-your-first-u-net-in-pytorch-b3297a844cf3>, 2023. [cit. 15. 5. 2024].
- [18] WANG, Y., LONG, M., WANG, J., GAO, Z., AND YU, P. S. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems* 30 (2017).
- [19] XAVIER, A. An introduction to convlstm. Dostupné na internete: <https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7/>, 2019. [cit. 15. 5. 2024].
- [20] ZHANG, R., ISOLA, P., EFROS, A. A., SHECHTMAN, E., AND WANG, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR* (2018).

# Prílohy

**Príloha A:** Všetky zdrojové kódy použité v tejto práci sú dostupné na GitHub-e na odkaze: <https://github.com/bszepesiova/ConvLSTM-Cloud-Prediction>