Using different Machine Learning Algorithms to predict Wildfires from spatial Climate Data in North America

Mikkel Knudsen, Andrea Vang, Svenja Frey | Applied Machine Learning | 12.06.2024

Using different Machine Learning Algorithms to predict Wildfires from spatial Climate Data in North America

Mikkel Knudsen, Andrea Vang, Svenja Frey | Applied Machine Learning | 12.06.2024

PROBLEM STATEMENT

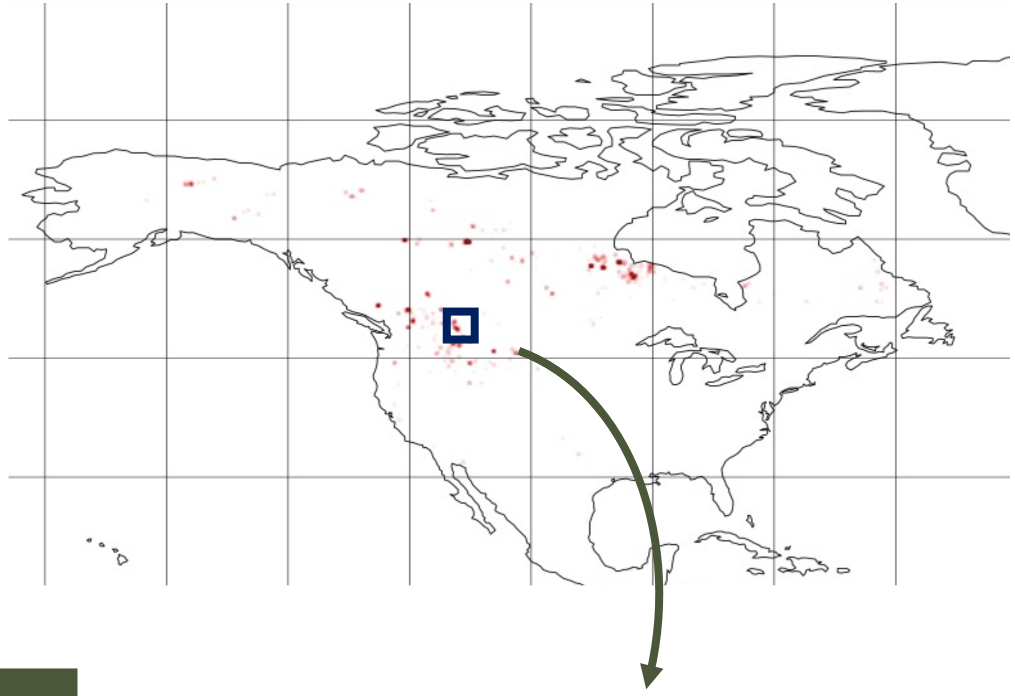
- Wildfires are increasing as climate change is progressing
- Need for new prediction methods to determine wildfire risk under climate change
- There might be a spatial dependency in the data

GOAL

Incorporate spatial aspects into the machine learning predictions based on climate data in North America.

PROJECT GOAL

- Predict the wildfire risk for each grid cell
- Use both XGBoost and CNN (Convolutional Neural Network)
- Tune the models to fit our climate data



Is there Fire?

GOAL
Incorporate spatial aspects into the machine learning predictions based on climate data in North America.

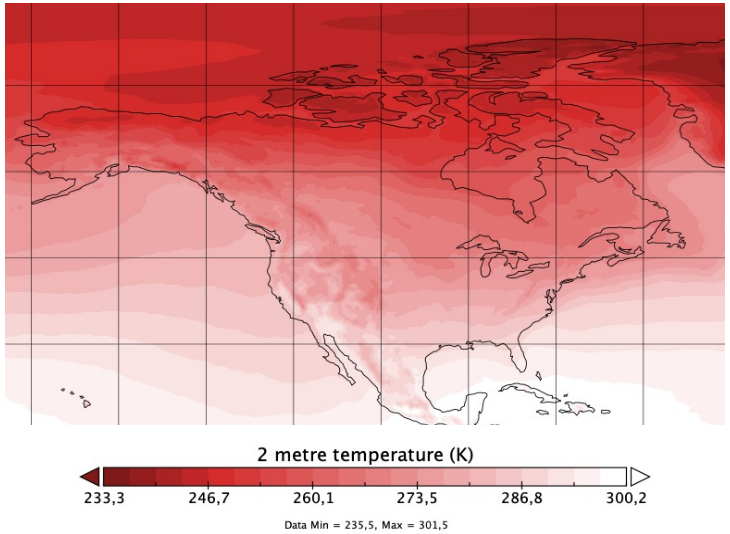
OUR DATA

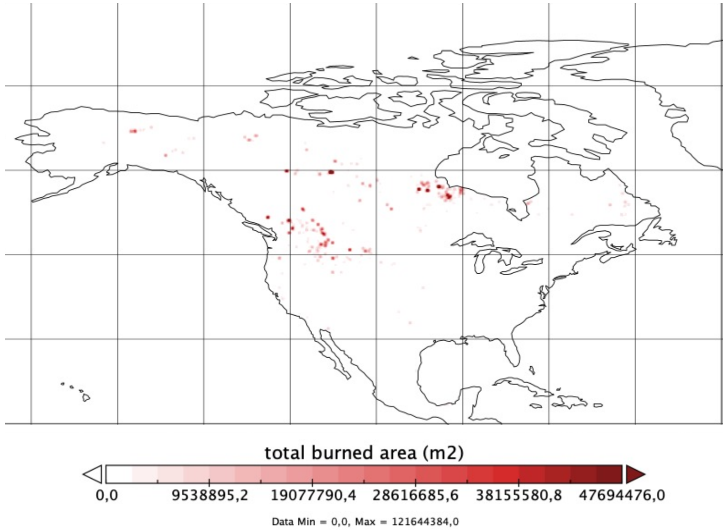
ERA5 reanalysis Dataset

- Reanalysis from CMIP5
- 0.25° x 0.25° grid
- Monthly data
- Many different climate variables → selection

Wildfire Data from Satellites

- Satellites → Gridpoints
- 0.25° x 0.25° grid
- Monthly data
- Target = burned area of grid cell





https://cds.climate.copernicus.eu/cdsapp#!/dataset/satellite-fire-burned-area?tab=form
https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels-monthly-means?tab=overview

OUR DATA – It's more than you think!

Size of Dataset

- 228 Months of Data (~ 10 years)
- 301 lat x 501 long (North America) = **150 801 grid points**
- **34 382 628 observations / feature**

Storage Problems

- On the Laptop
- For RAM (e.g. Google Colab)

Downsizing

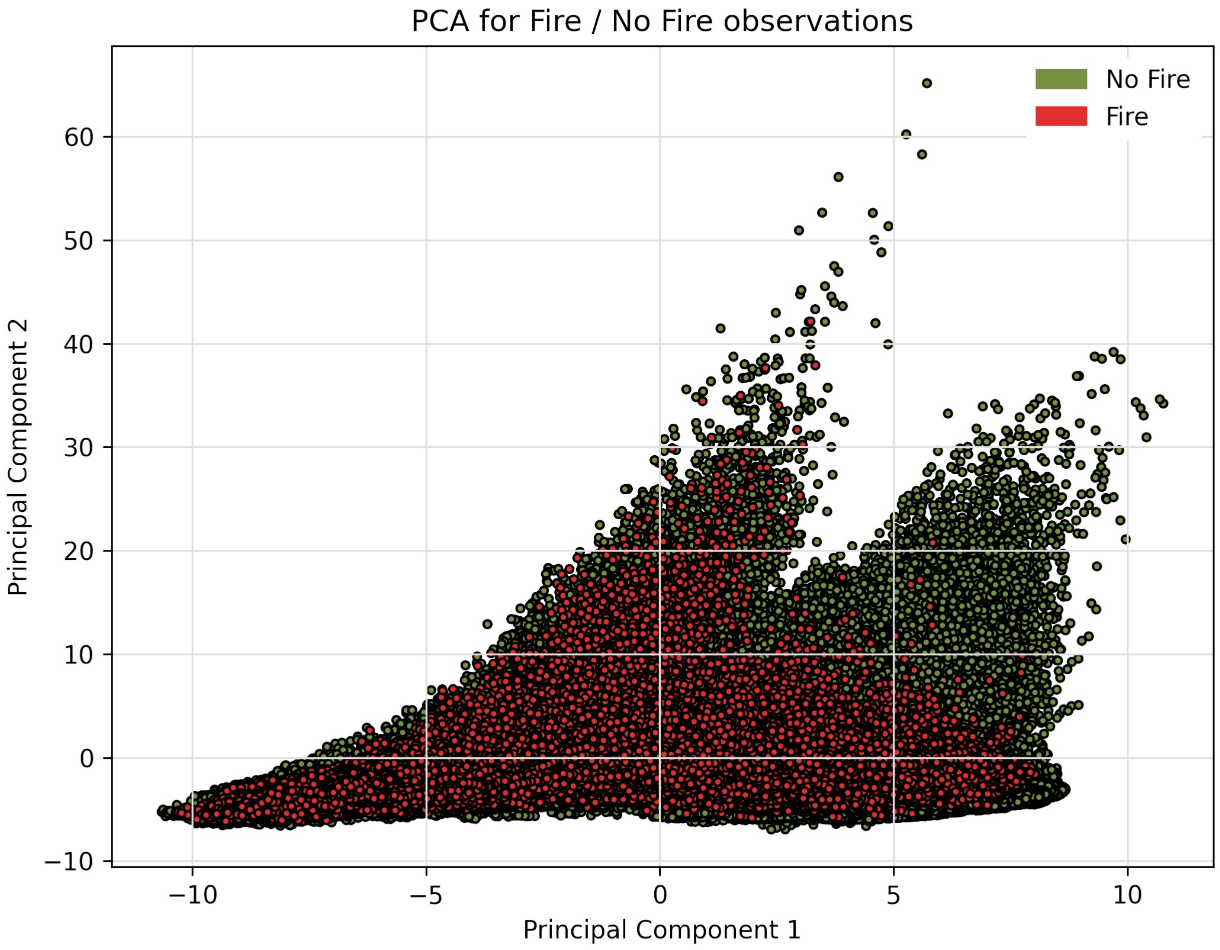
- Seasonal & Climate Variability needs to be preserved!

› It is difficult to downsize the problem. We decided to select the warmer months April – October.

UNDERSTANDING THE DATA

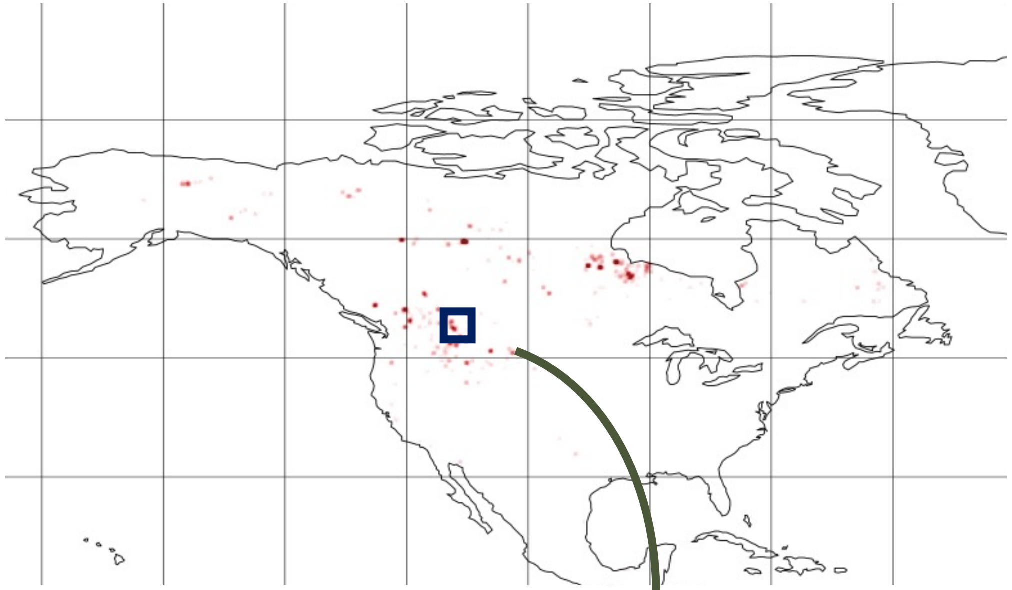
IS THERE A FIRE?

- Difficult to differentiate between Fire and No Fire Data Points
- Only first 2 PCAs → Many features / variance probably not captured in Figure
- Human impact on Wildfires



A (SIMPLE) MODEL

- Every gridpoint has a value for each feature
- Solution: Tabulate the data – each row is a gridpoint, each column is a feature
- Use burned area as target variable and apply ML model (XGBoost)

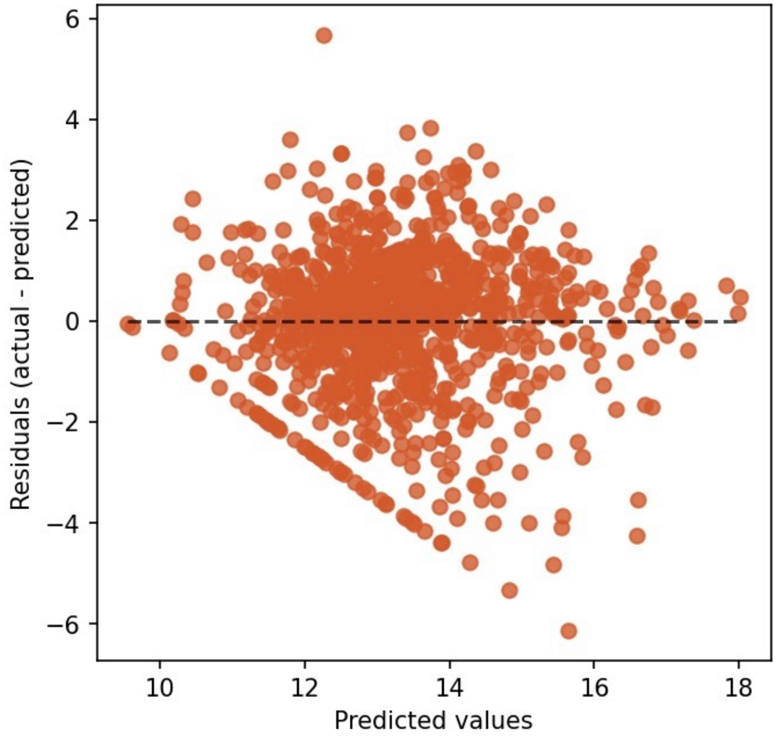


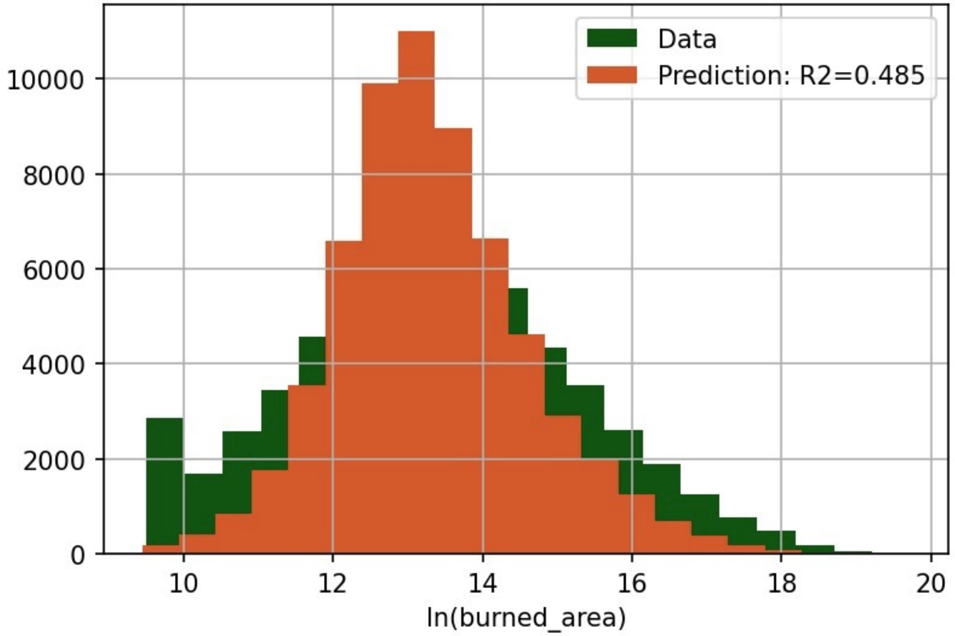
u10	v10	si10	t2m	cdir	cbh	cp
-0.947506	-4.0055785	7.0724893	265.42358	15759183.0	285.30063	5.53055e-05
0.6247316	-2.8558722	5.8690624	269.9707	21796160.0	388.92175	3.2355598e-05
0.88534915	0.9009036	4.9620504	278.1036	24560082.0	511.8433	0.00015726326
0.86207974	0.24246149	4.759609	280.75696	22540692.0	291.78683	8.76611e-05
1.0424178	-0.88629645	5.5724134	280.66397	17142272.0	364.24252	0.00050941255
-0.13074912	-1.0087018	5.3783875	277.78796	10336863.0	514.6909	0.00021256876
-1.5439789	-4.2076707	6.111945	270.63205	4384215.0	726.83734	0.000117383104

A (SIMPLE) REGRESSION MODEL

… and the reason we decided to do classification instead

- XGBoost | Regression | 50 Features
- Logarithmic Scaling of Target Feature



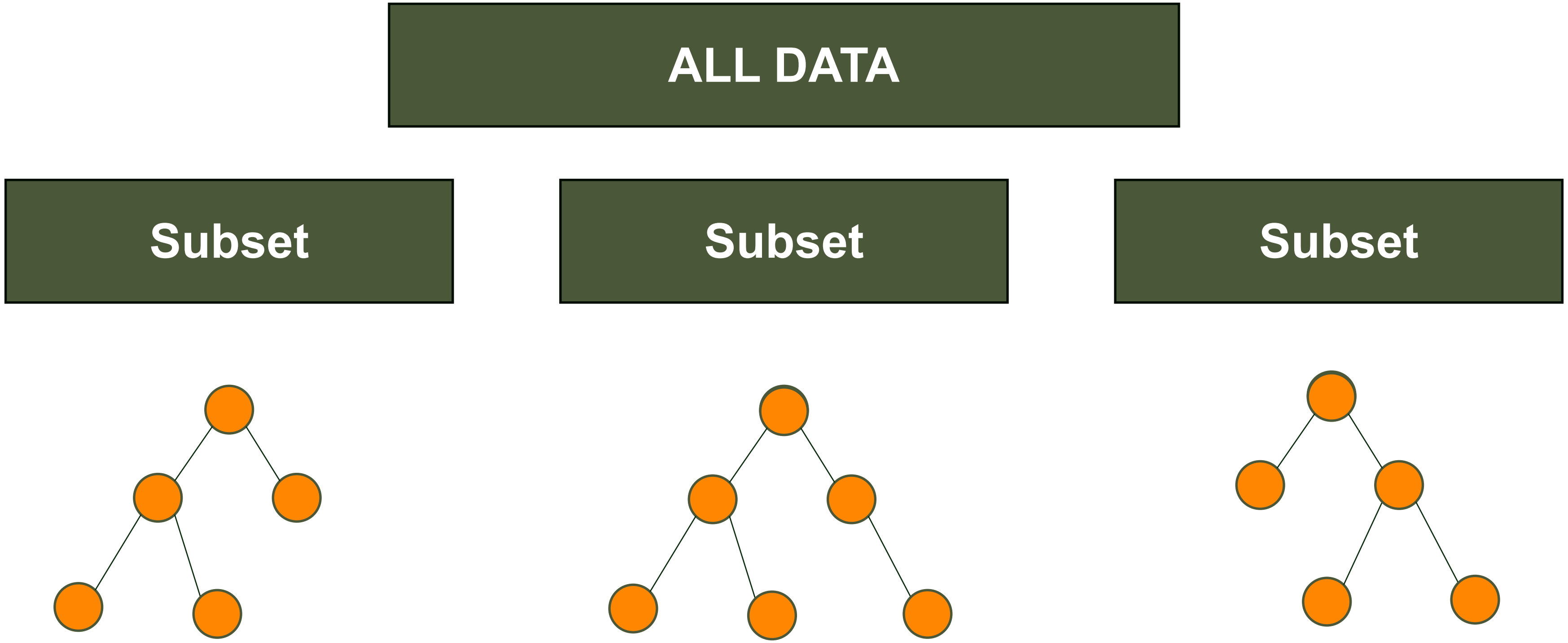


CLASSIFICATION

It was difficult to sample the tails of the "burned area" distribution, but the model was able to detect wildfires → switch to Classification

A (SIMPLE) CLASSIFICATION MODEL

- XGBoost | Classification | Feature-Selection | Class Weighting

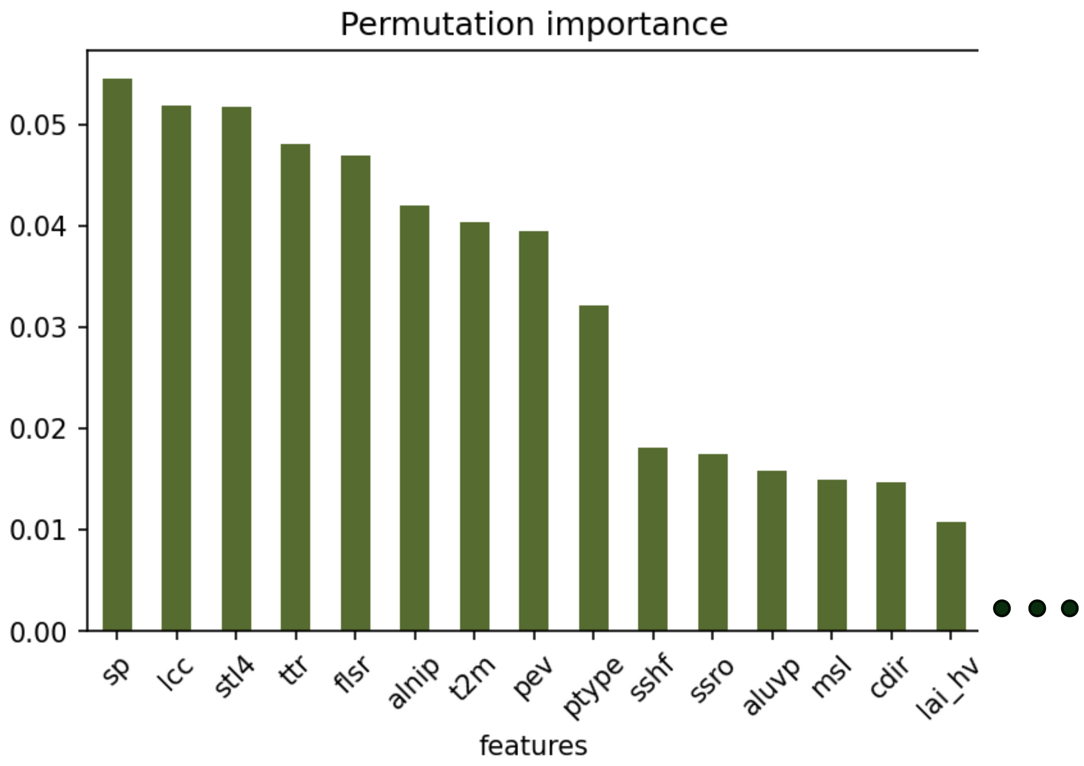


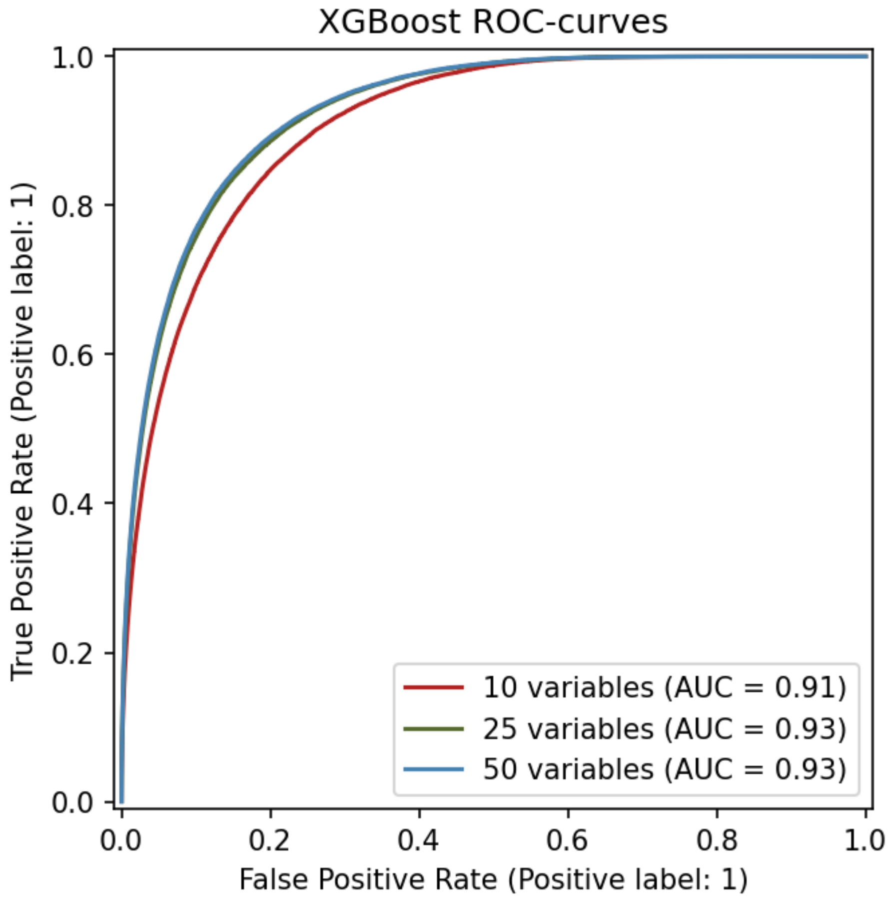
Disclamer: Model for Illustration Purposes

A (SIMPLE) CLASSIFICATION MODEL

… Feature Selection

- XGBoost | Classification | 25 Features + SST





A (SIMPLE) CLASSIFICATION MODEL

… Class Weighting

[%]	Predicted Label 0	Predicted Label 1
True Label 0	95	0.46
True Label 1	3.4	**1.2**

No weighting

[%]	Predicted Label 0	Predicted Label 1
True Label 0	94	1.6
True Label 1	2.5	**2.1**

Balanced weighting

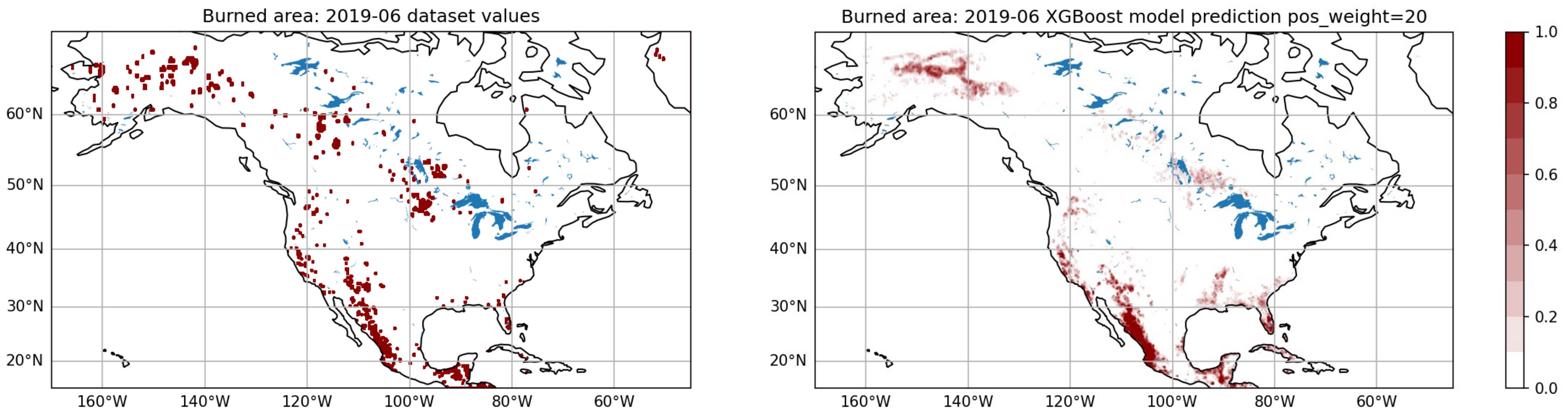
UNBALANCED DATASETS

- Datapoints no Fire **>>** Datapoints Fire
- The model predicts more false negatives
- Punish false negatives more by weigthing the classes

› The model improved when it learned on all data points and fires were weigthed higher.

A (SIMPLE) CLASSIFICATION MODEL

- XGBoost | Classification | 25 Features

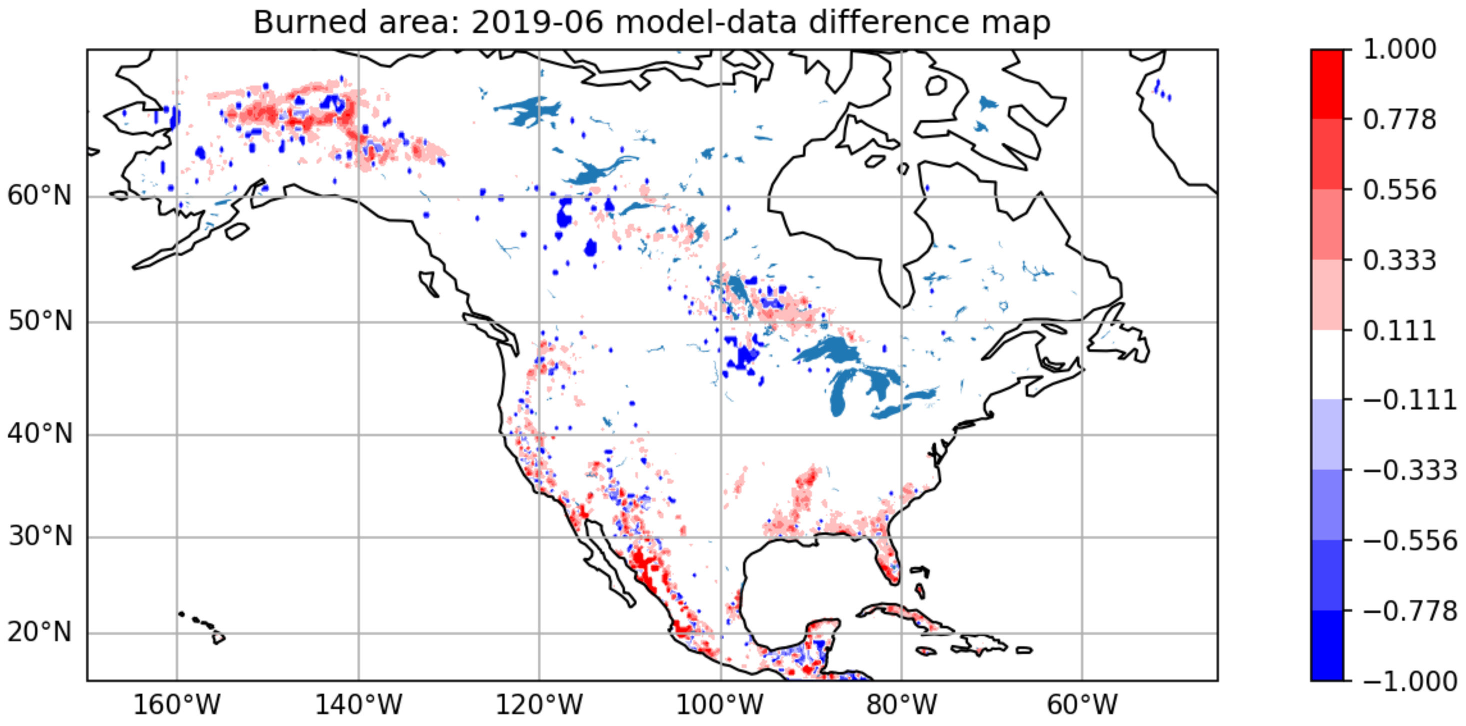


› Feature Selection
› Weighting of Classes

› Accuracy = 96.6 %
› Log-loss = 0.08512

A (SIMPLE) CLASSIFICATION MODEL

- XGBoost | Classification | 25 Features

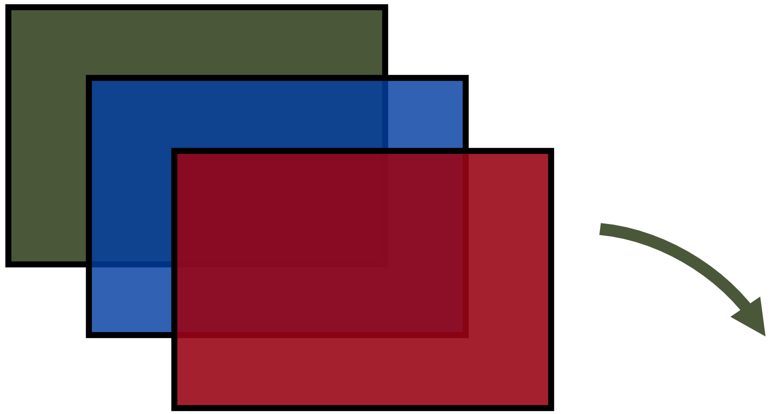


› Feature Selection
› Weighting of Classes

› Accuracy = 96.6 %
› Log-loss = 0.08512

THE SPATIAL MODEL – CNN

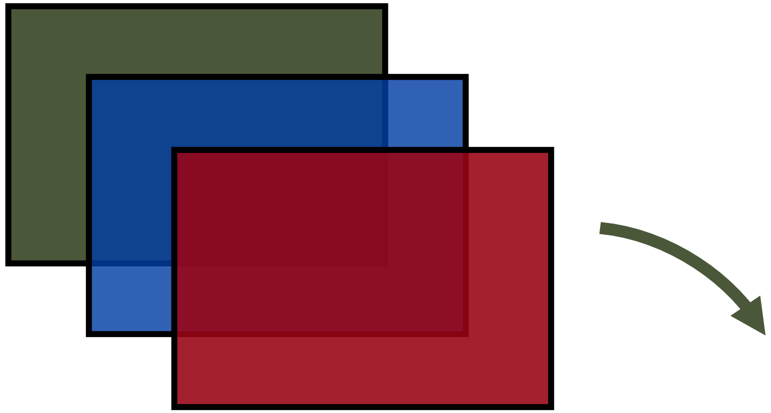
- **CNN** = **C**onvolutional **N**eural **N**etwork
- Introduces spatial dependencies to the model



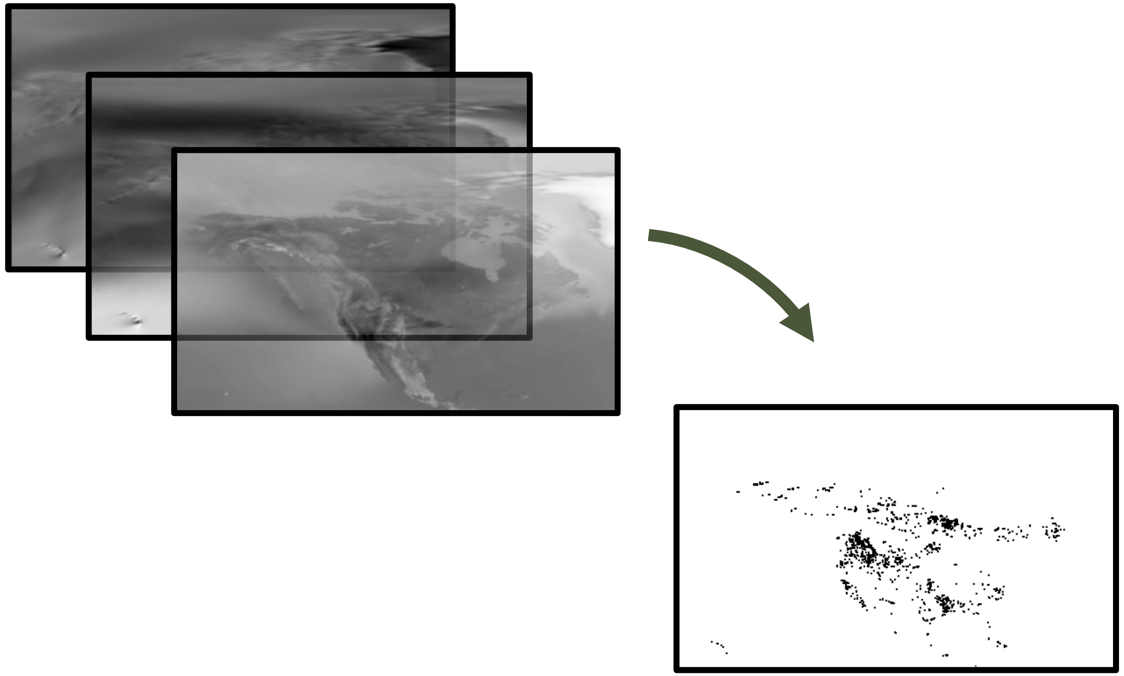
Cat or Dog?

THE SPATIAL MODEL – CNN

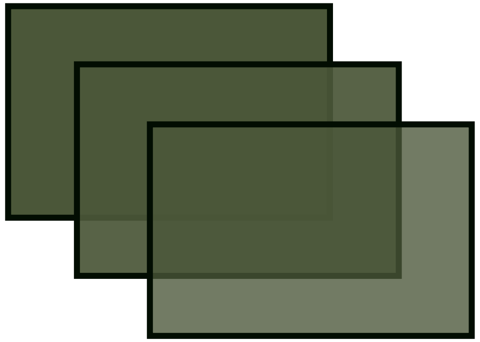
- **CNN** = **C**onvolutional **N**eural **N**etwork
- Introduces spatial dependencies to the model

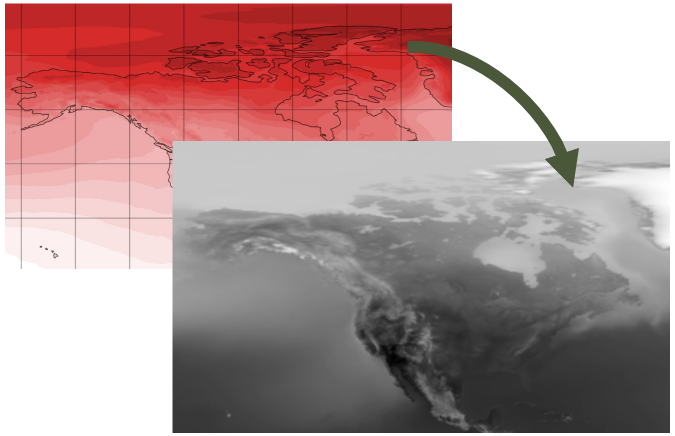


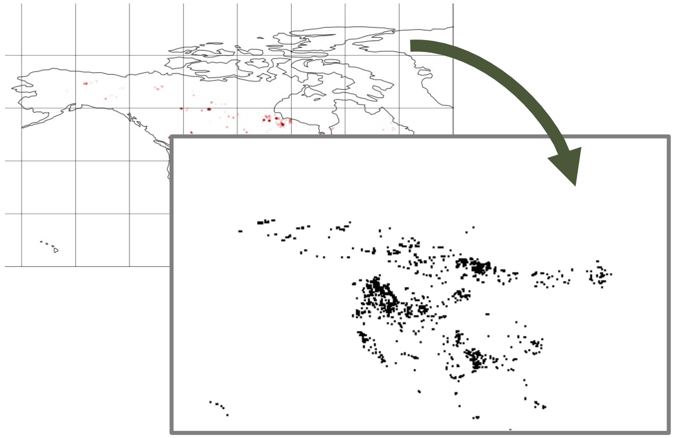
Cat or Dog?



THE SPATIAL MODEL – CNN SETUP







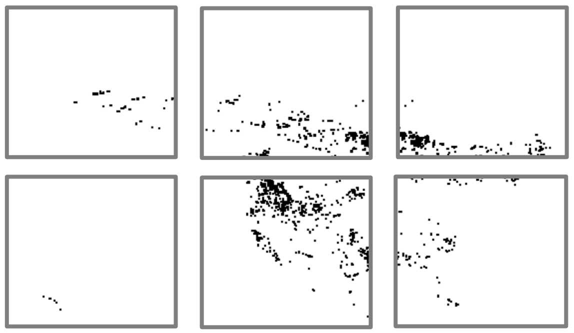
Making Pictures
Translating the grid point data into .png format

Grayscale of Features
Transform the data into grayscale images to reduce data size

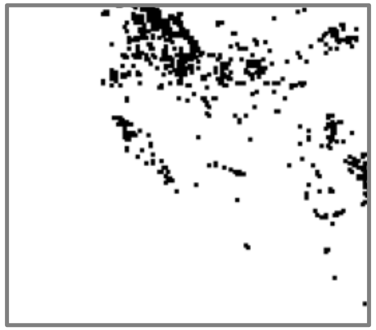
Classification of Target
Using binary classification instead of regression

› CNN is more complex in the setup process, since the spatial aspects need to be preserved.

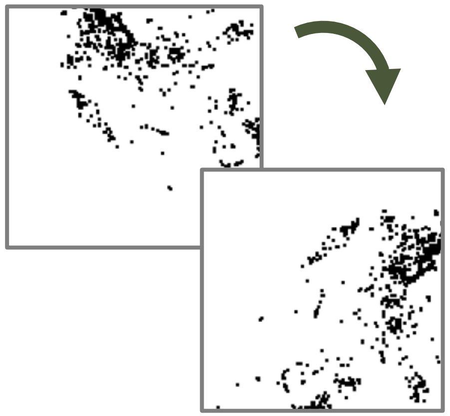
THE SPATIAL MODEL – CNN SETUP



Splitting Pictures
Making smaller tiles to focus on areas with more fire.



Selecting Tiles
Balancing dataset in terms of fires / no fires



Data Augmentation
Rotating, mirroring, ….

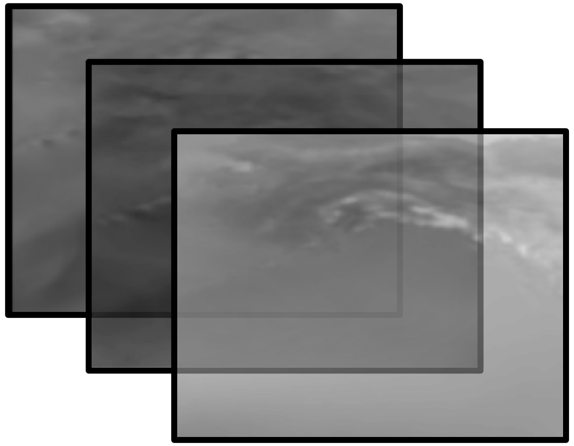
› An unbalanced dataset concerning the target variable is quite tricky for a CNN.

THE SPATIAL MODEL – CNN

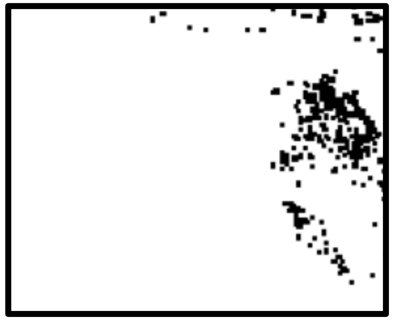
› **CNN** = **C**onvolutional **N**eural **N**etwork

› Introduces spatial dependencies to the model

› Pixel-by-Pixel prediction



CNN



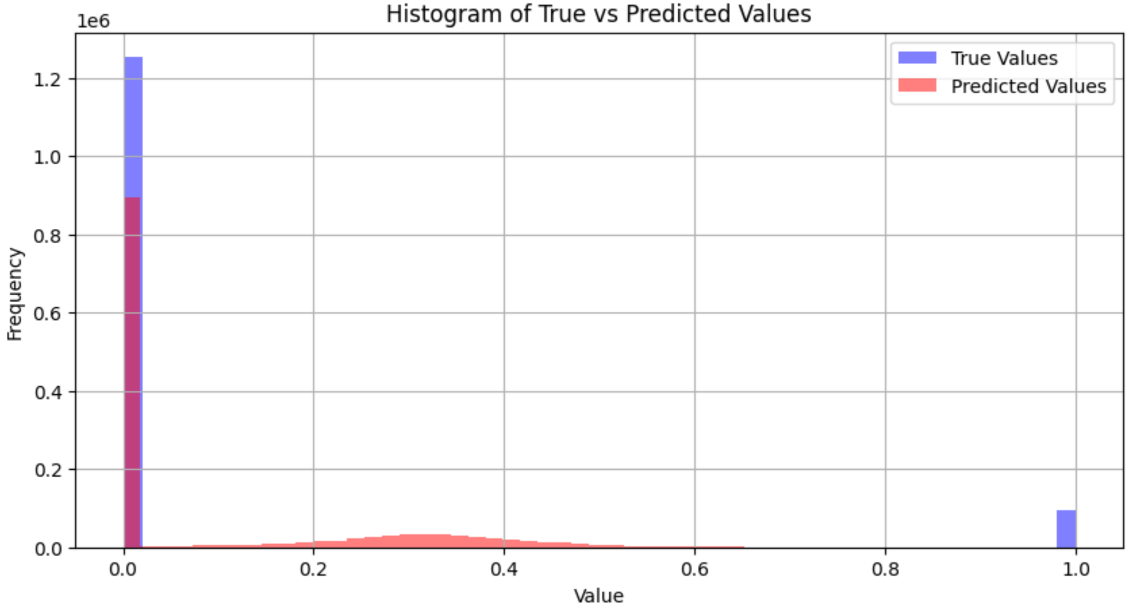
Input dimensions:
n **x** 8 pixel **x** 8 pixel **x** 26 features

Output dimensions:
n **x** 8 pixel **x** 8 pixel **x** 1

n = # samples

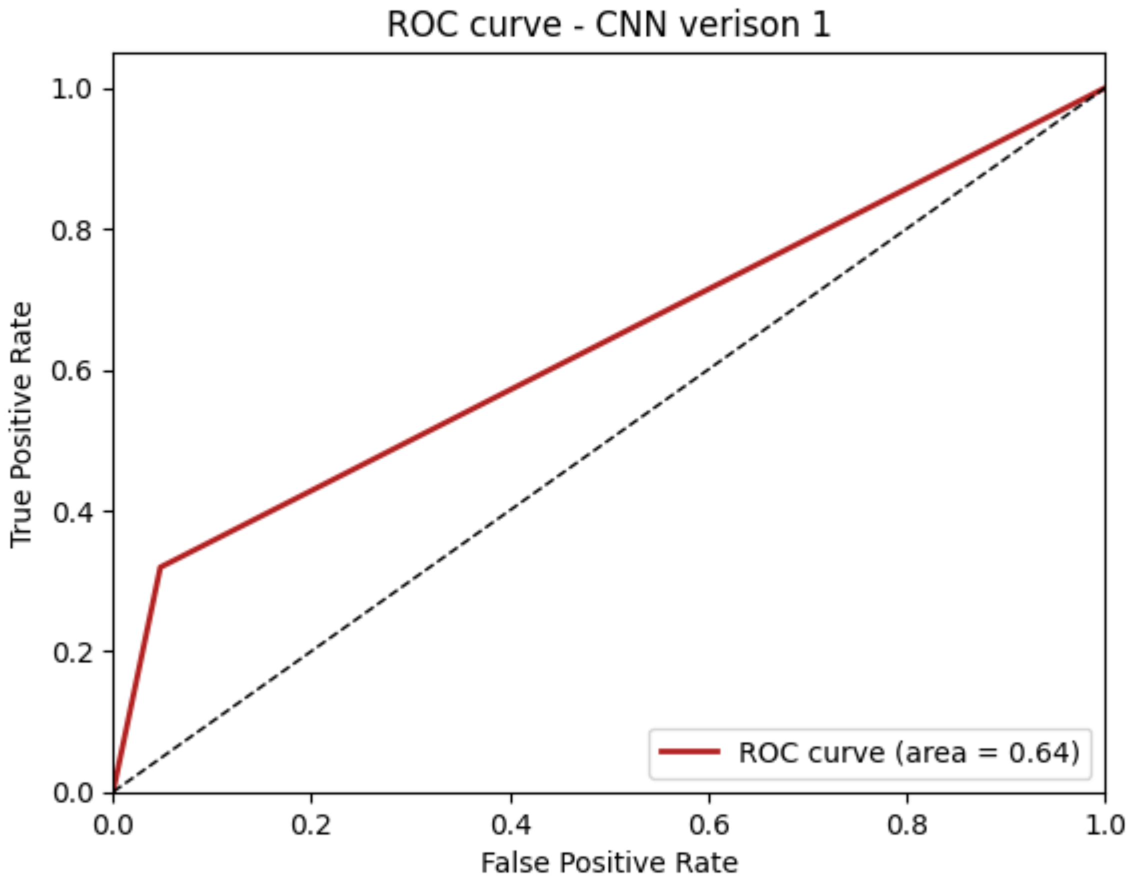
CNN CHALLENGES

- Balancing the predicted values was tricky
- And the model is bad :')



0 = no fire

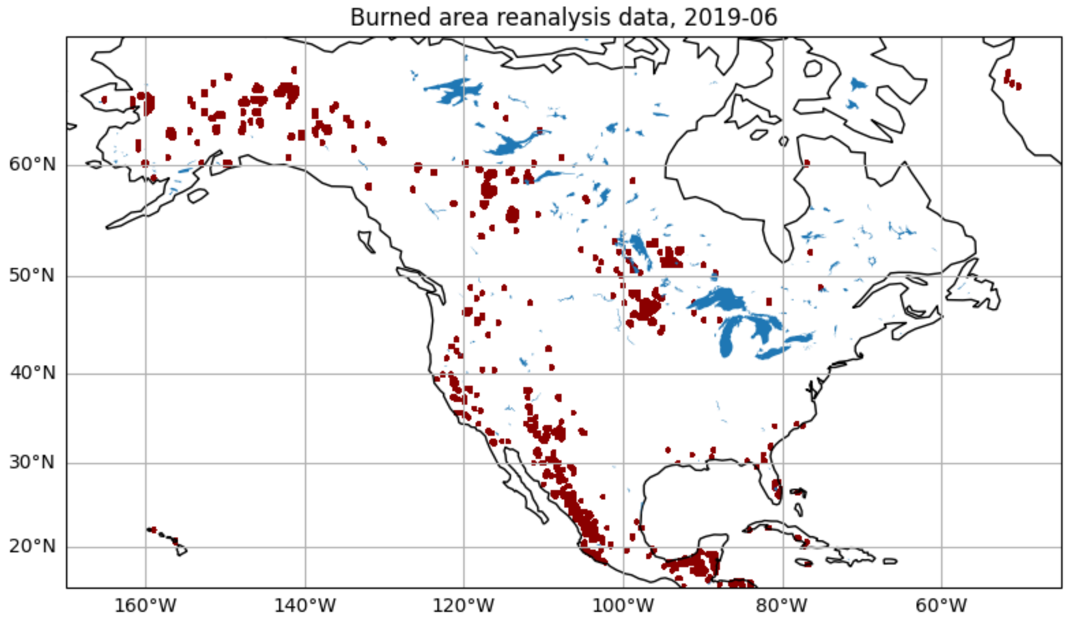
1 = fire



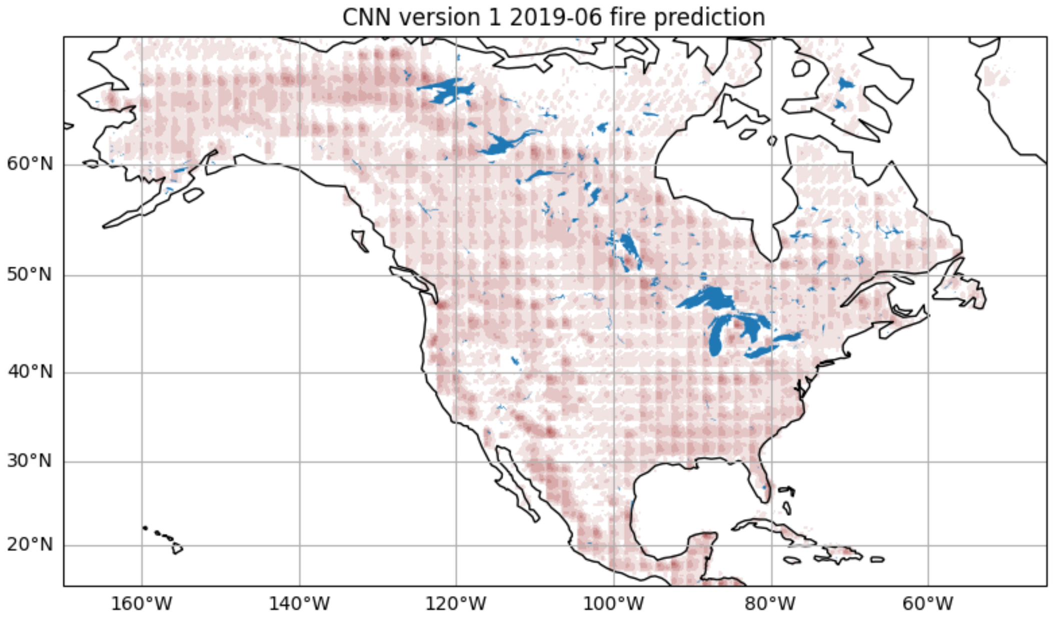
CNN CHALLENGES

› Our initial try was to predict which pixels in an 8x8 tile were burning using a CNN

› We realised that this kind of generative CNN may not be optimal



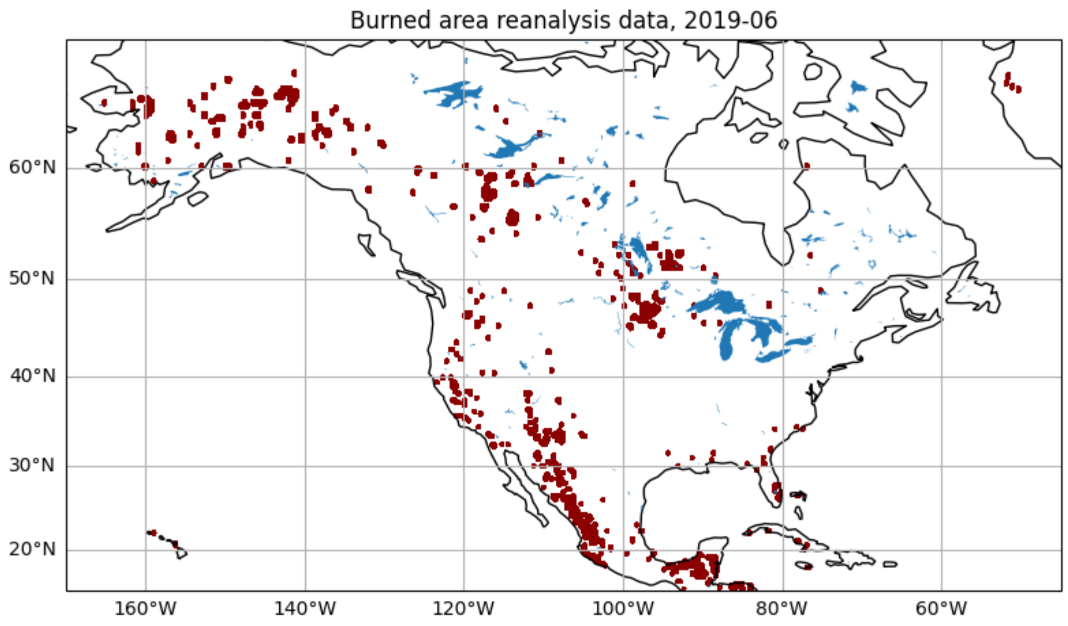
True Fire



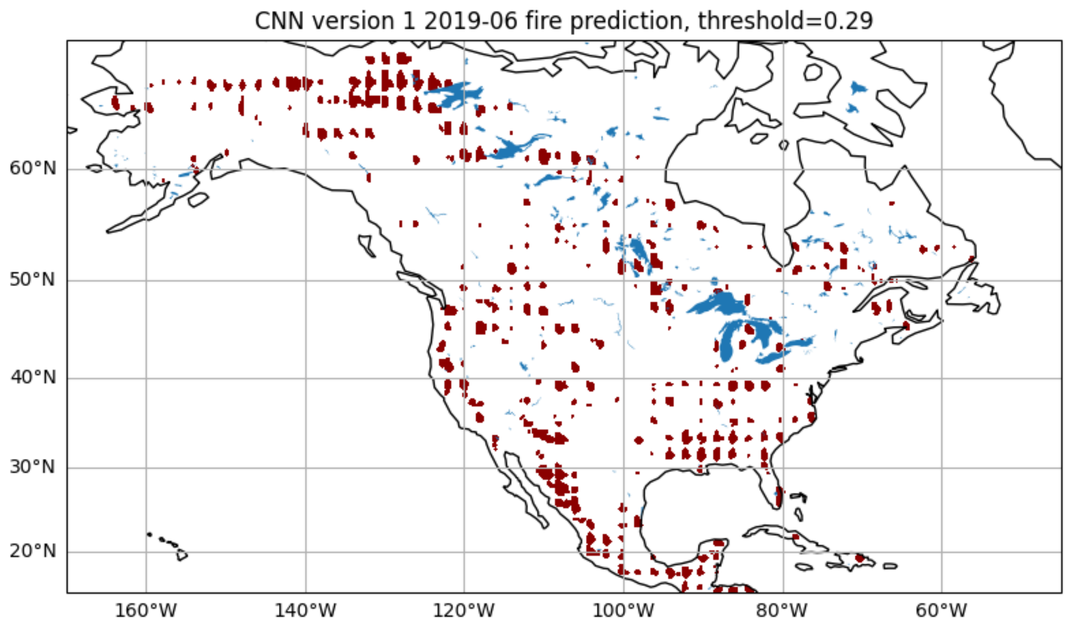
Predicted Fire

CNN CHALLENGES w Version 1

- Our initial try was to predict which pixels in an 8x8 tile were burning using a CNN
- We realised that this kind of generative CNN may not be optimal



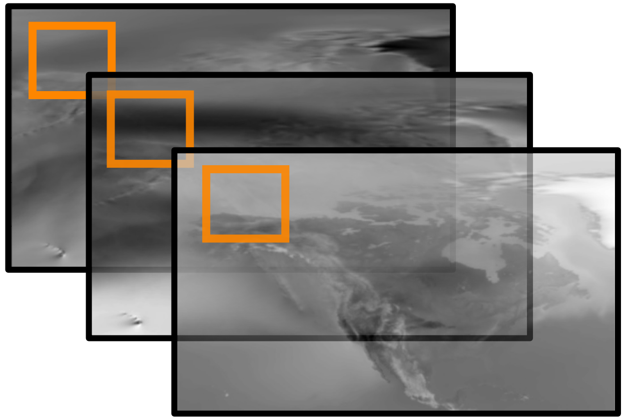
True Fire



Predicted Fire, threshold of 0.29

THE SPATIAL MODEL – CNN Version 2

- **CNN** = **C**onvolutional **N**eural **N**etwork
- Make 1 Classification for a 3x3 grid, predicting the fire for the central grid point



CNN

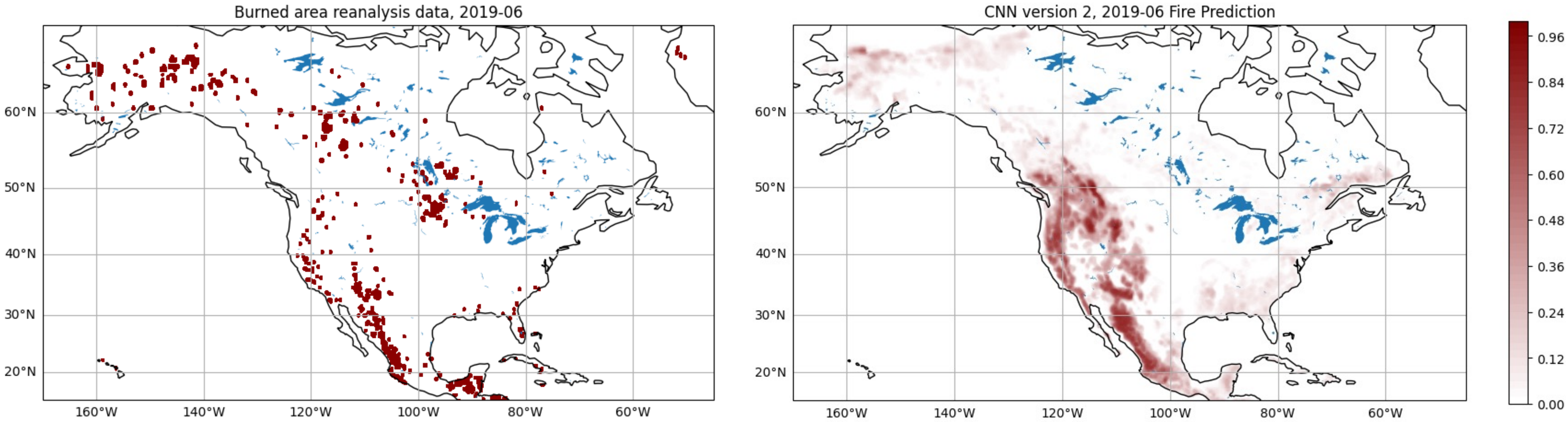
Fire Prediction
Classification in [0,1]

Input dimensions:
n **x** 3 pixel **x** 3 pixel **x** 26 features

Output dimensions:
n **x** 1

n = # samples

THE SPATIAL MODEL – CNN Version 2

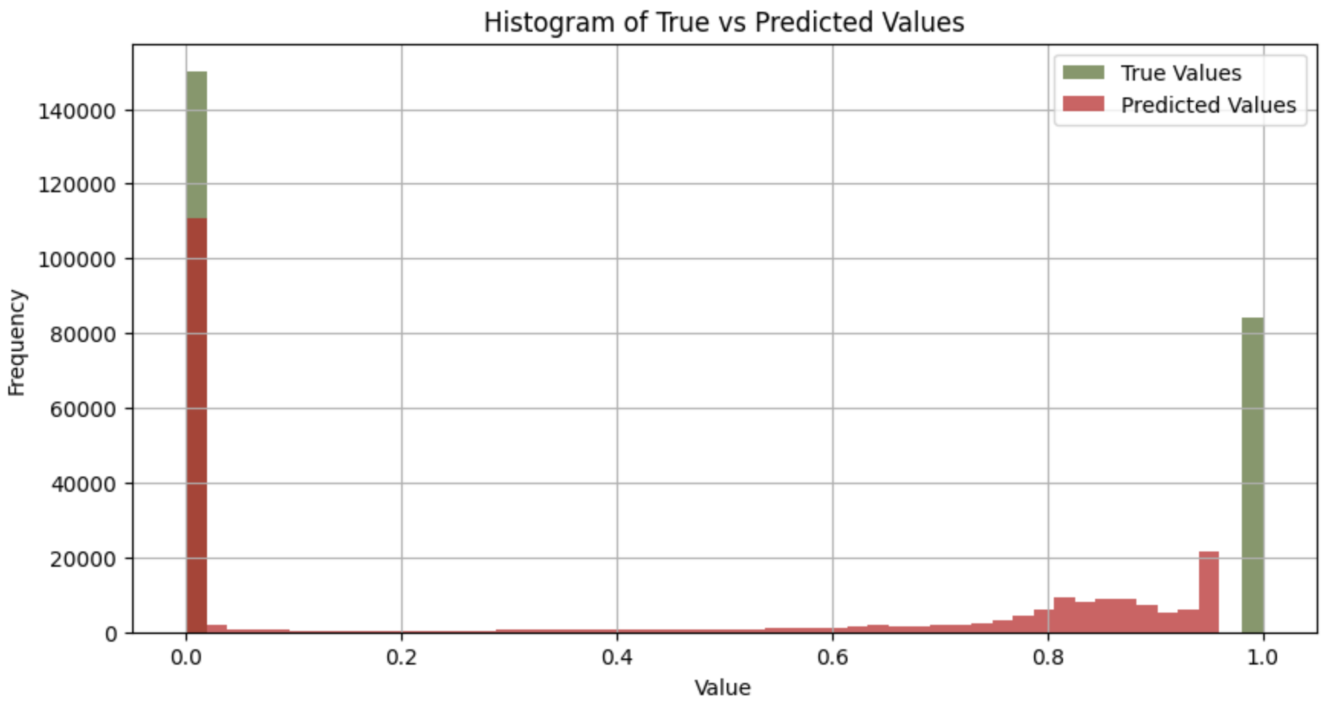


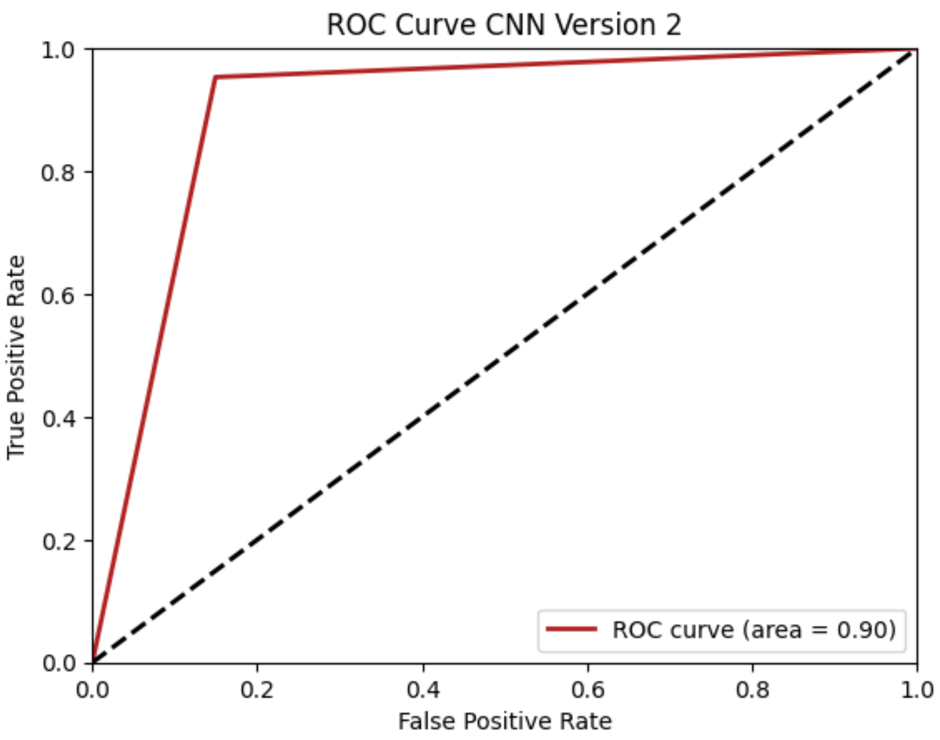
True Fire

Predicted Fire

- Model can capture some of the spatial patterns of the wildfire
- Some areas are overrepresented / underrepresented

THE SPATIAL MODEL – CNN Version 2





- Separation of fire / no fire in the predicted values
- Clear threshold value based on ROC curve (AUC = 0.90)

LIMITATIONS

Our wildfire prediction models have some limitations in their application. Most importantly, they are limited to feature data that is in the style of the ERA5 dataset.

Temporal Aspects

- Limited by data availability, RAM & complexity of GRU setup

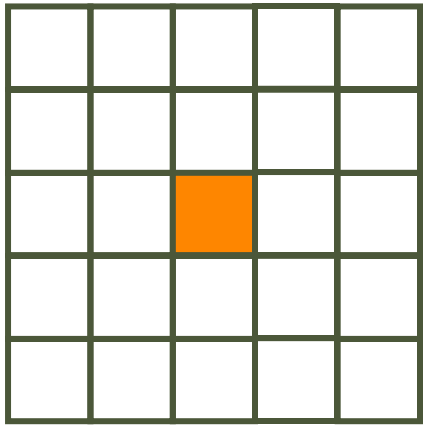
Data Limitations

- Data Scale limits Predictions Scale; Many features used limits real world application

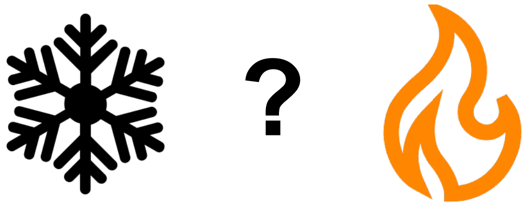
Human Impact

- Humans heavily impact wildfires (90%; Liz-Lopez, 2024); not accounted for in our model

OUTLOOK



5x5 Scanner
Use CNN Version 2 with larger grid pictures.
(RAM Problems)



Yearly Predictions
Make predictions on Winter Months

Future Predictions
Apply the model on future climate predictions

- There are many more applications for our Fire Prediction Models, especially locally!

WHAT TO LEARN FROM US

Unbalanced data is tricky

Wildfire prediction is difficult

XGBoost is awesome

Be specific about research method

CNNs are not always awesome

THANKS FOR LISTENING!

REFERENCES

- Helena Liz-López, Javier Huertas-Tato, Jorge Pérez-Aracil, Carlos Casanova-Mateo, Julia Sanz-Justo, David Camacho, *Spain on fire: A novel wildfire risk assessment model based on image satellite processing and atmospheric information*, Knowledge-Based Systems, Volume 283, 2024, 111198, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2023.111198.
- S. Liu, H. Ji and M. C. Wang, "Nonpooling Convolutional Neural Network Forecasting for Seasonal Time Series With Trends," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 8, pp. 2879-2888, Aug. 2020, doi:10.1109/TNNLS.2019.2934110.
- Fire data: https://cds.climate.copernicus.eu/cdsapp#!/dataset/satellite-fire-burned-area?tab=form
- Reanalysis data: https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels-monthly-means?tab=overview

APPENDIX

DATA PREPROCESSING

Data preprocessing included not only the steps below, but also handling the data in terms of file format, dimension of data for the models and converting data to images (.png)

Regridding

- Transform both datasets to the same grid to make sure gridpoints overlap

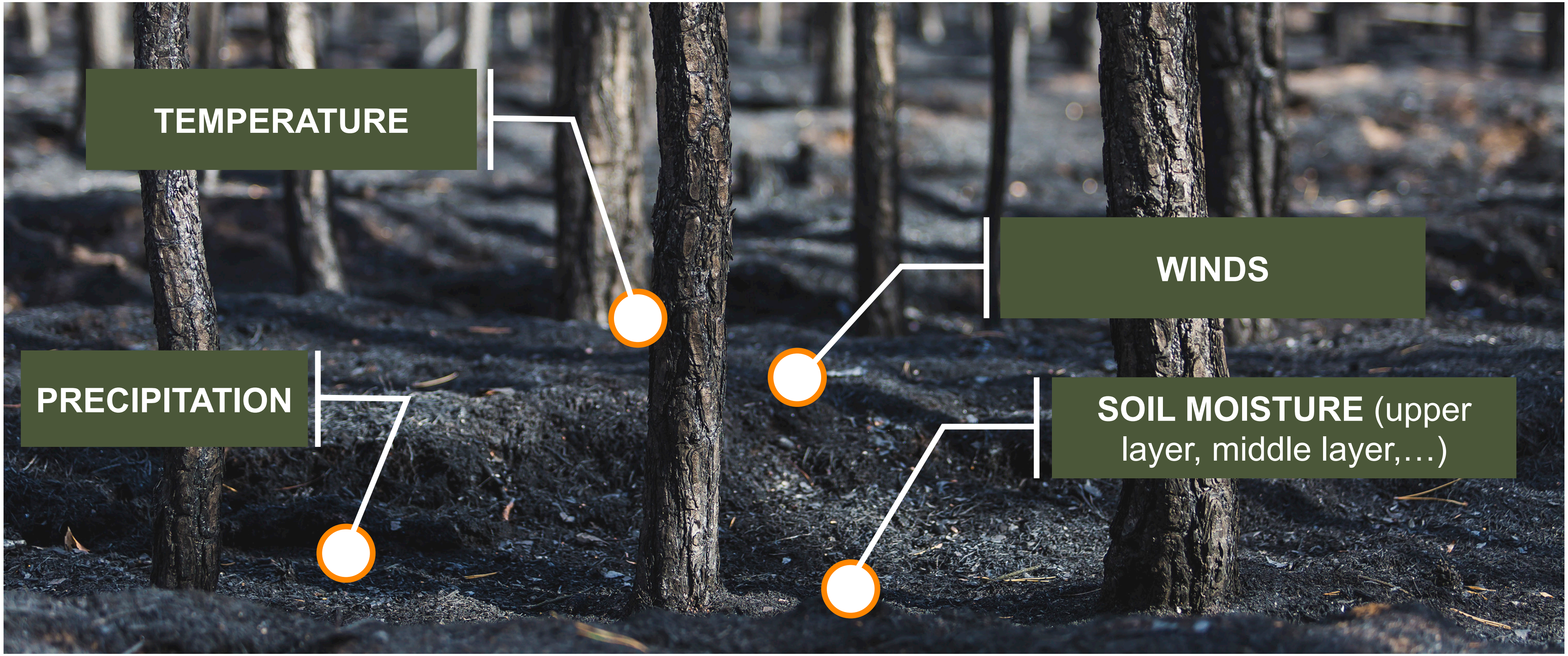
Standardizing

- Normalize and center the data to introduce consistency concerning different units

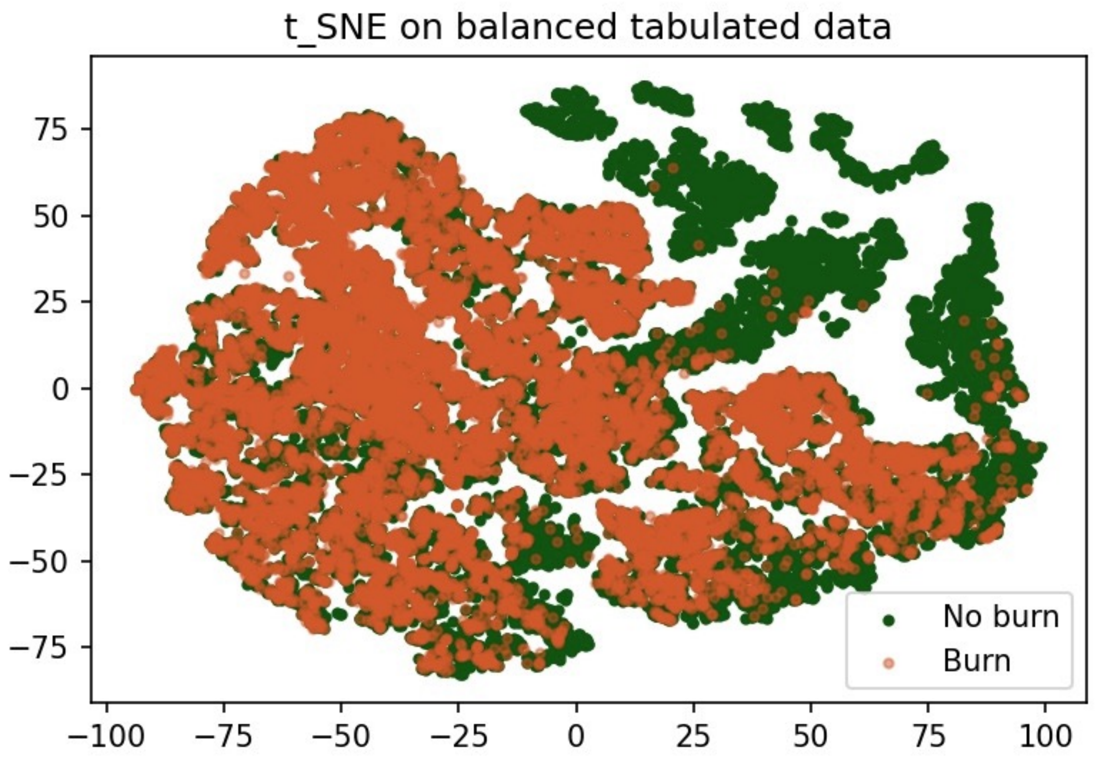
Classification

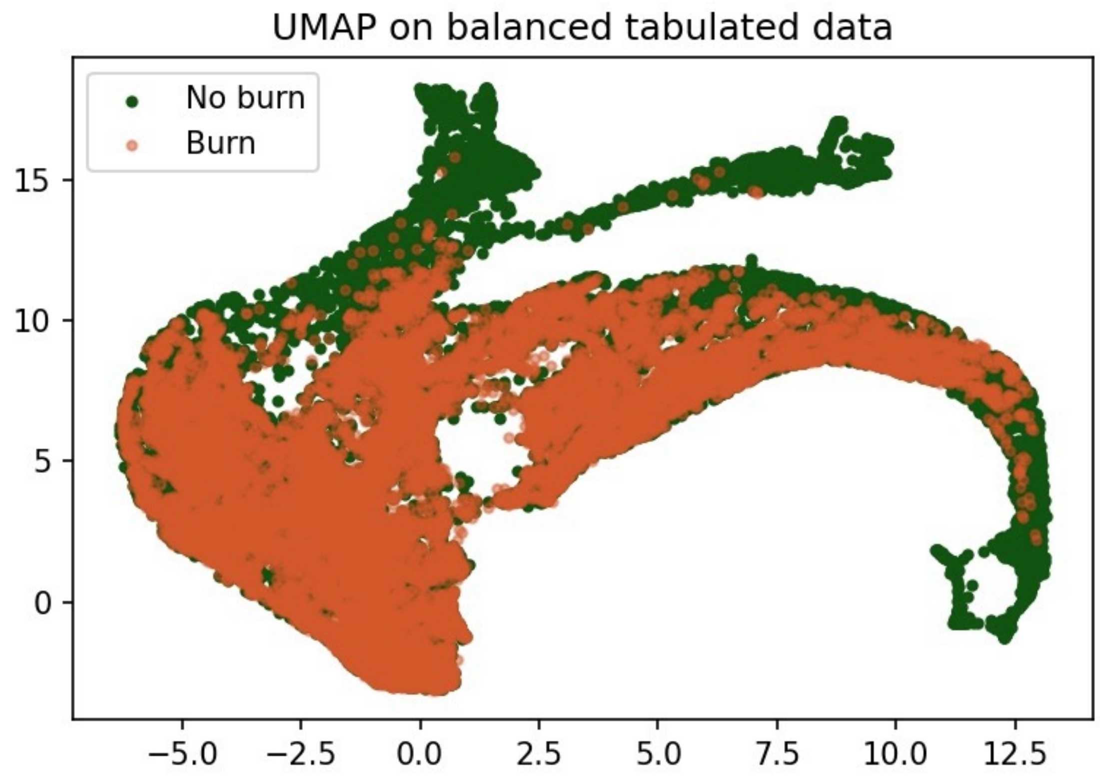
- Transform a continours variable (burned area) to a binary classification (0 = not burned, 1 = burned)

FEATURE SELECTION



UNDERSTANDING THE DATA





FEATURE SELECTION - XGBoost

We can make a feature selection based on domain knowledge, but still ran the models with a high number of features as we were surprised by which ones are important to the model!

Correlation between features

- We omitted features that had a very high correlation with others.

Permutation results

- Features were selected based on Permutation Results from the Models.

Latitude & Longitude information

- We tested their impact on the model → The model learned on them quite a bit!

A (SIMPLE) CLASSIFICATION MODEL

… Feature Selection

- XGBoost | Classification | 25 Features
- \+ Longitude and Latitude as Features
- Improved accuracy: ~1 %

LONGITUDE & LATITUDE

- Measurement location is likely known
- Fires often occur at the same place
- We didn’t want the model to predict based on that, as it would have problems identifying new fire locations

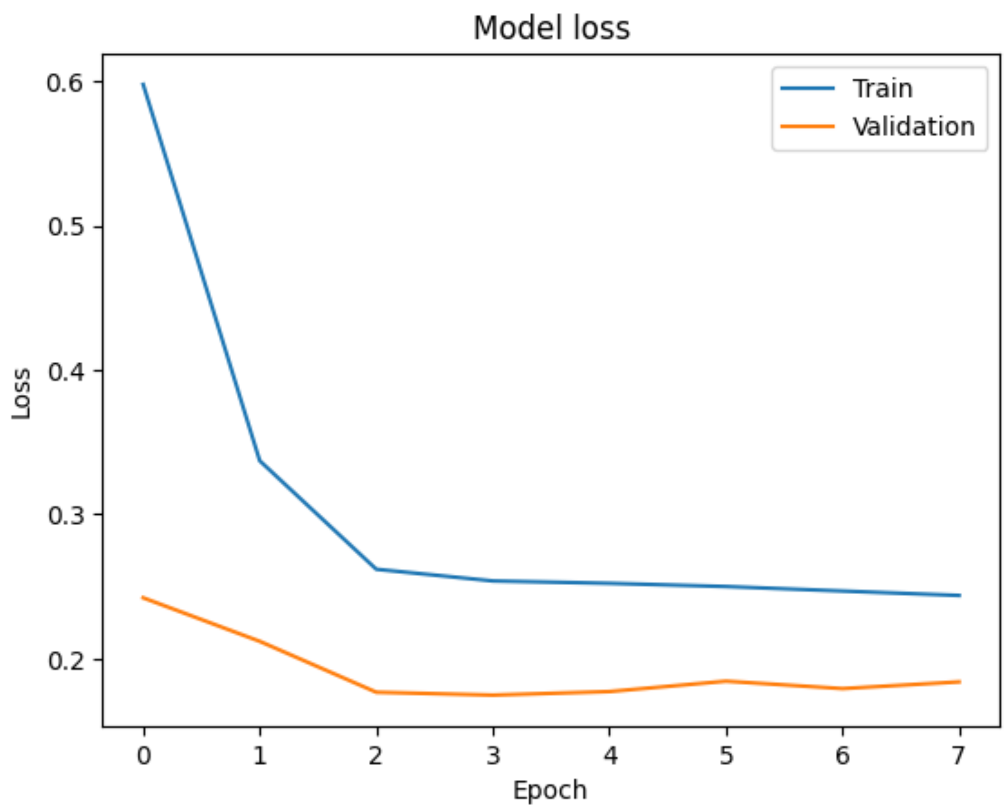
A (SIMPLE) CLASSIFICATION MODEL

… Final Hyperparameters

- xgb.XGBClassifier(n_estimators=150,
- max_depth=30,
- max_bin=100,
- learning_rate=0.05,
- tree_method="hist",
- scale_pos_weight=20,
- early_stopping_rounds=2,
- objective='binary:logistic')

THE SPATIAL MODEL – CNN Version 1

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 8, 8, 4)           940

 conv2d_10 (Conv2D)          (None, 8, 8, 4)           148

 conv2d_11 (Conv2D)          (None, 8, 8, 4)           148

 flatten_3 (Flatten)         (None, 256)               0

 dense_3 (Dense)             (None, 256)               65792

 reshape_2 (Reshape)         (None, 8, 8, 4)           0

 conv2d_transpose_3 (Conv2D  (None, 8, 8, 1)           37
 Transpose)

=================================================================
Total params: 67065 (261.97 KB)
Trainable params: 67065 (261.97 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



› Here is the model structure and the loss

THE SPATIAL MODEL – CNN Version 1

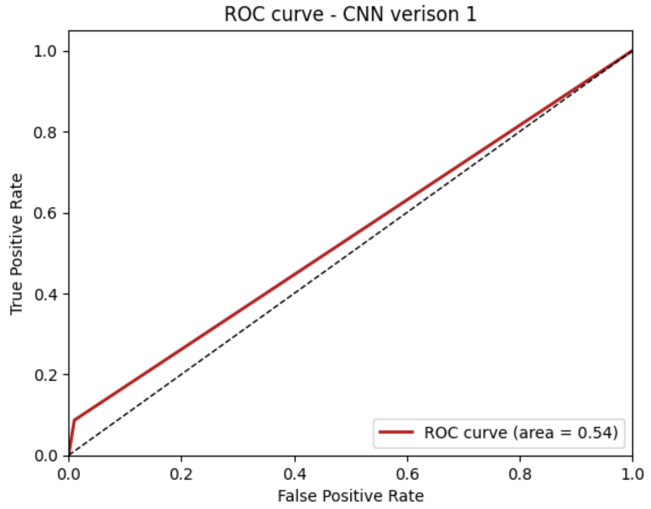
```
from sklearn.model_selection import train_test_split
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.3, random_state=39)
```

```
# ModelCheckpoint callback to save the best model based on validation loss
checkpoint = ModelCheckpoint('best_model.h5', monitor='val_loss', save_best_only=True, mode='min')

# EarlyStopping callback to stop training when validation loss stops improving
early_stopping = EarlyStopping(monitor='val_loss', patience=4, mode='min', restore_best_weights=True)


class_weight = {0: 1.0, 1: 2.0}  # Weigths of classes

history = model.fit(
    X_train, y_train, batch_size=32,
    epochs=200,
    validation_data=(X_val, y_val),
    class_weight=class_weight,
    callbacks=[checkpoint, early_stopping]
)
```

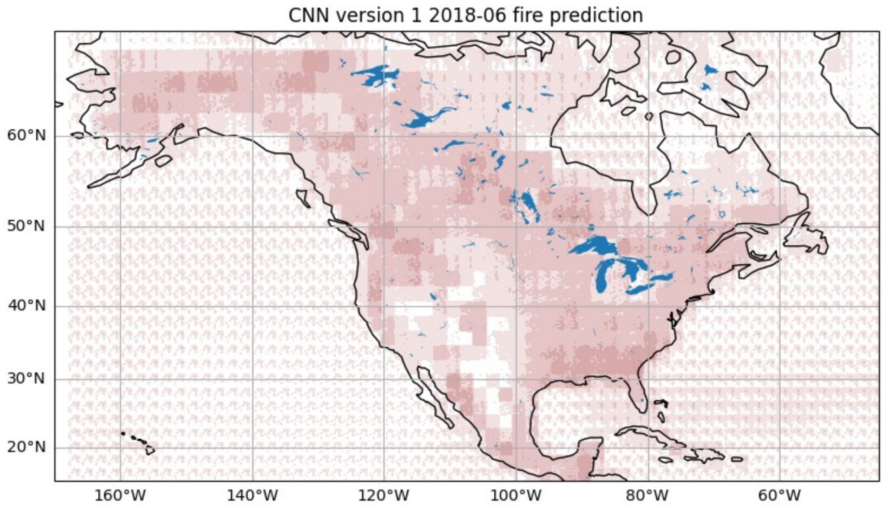
- Splitting and hyperparameters
- We used, relu activation, padding, 3x3 kernel, the optimizer was adam and loss was binary crossentropy

THE SPATIAL MODEL – CNN Version 1

- We ended up not using data augmentation for this version of the CNN, since it actually made
  the model perform visually worse. We know that this reduces the general applicability of the model.
- The ROC curve to the right shows that the model performed worse than the one w.o. data augmentation.
- We are not exactly sure why, but believe it has something to do with overfitting, as it predicts the same shape in each
  tile just with different intensities, although we use a very simple model and few iterations.
- This might also be because in general CNNs are better for classifying a picture, than for generating or predicting a picture.

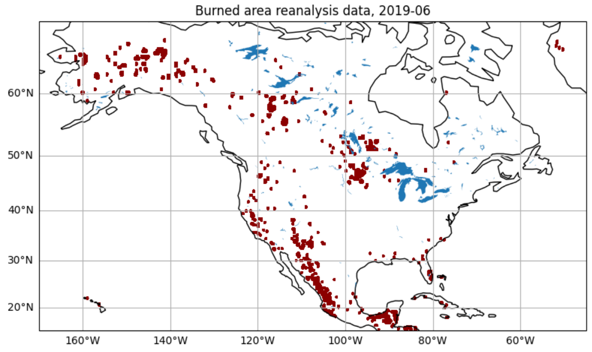


- ROC curve
  including data augmentation

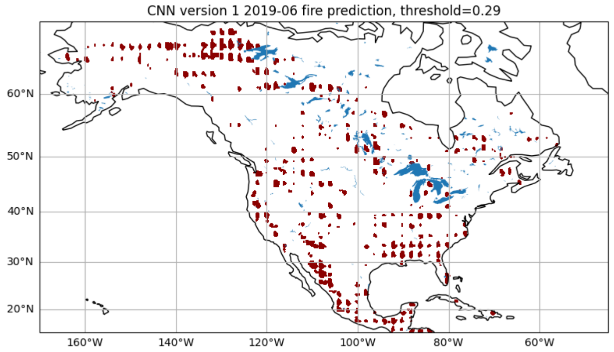


CNN CHALLENGES

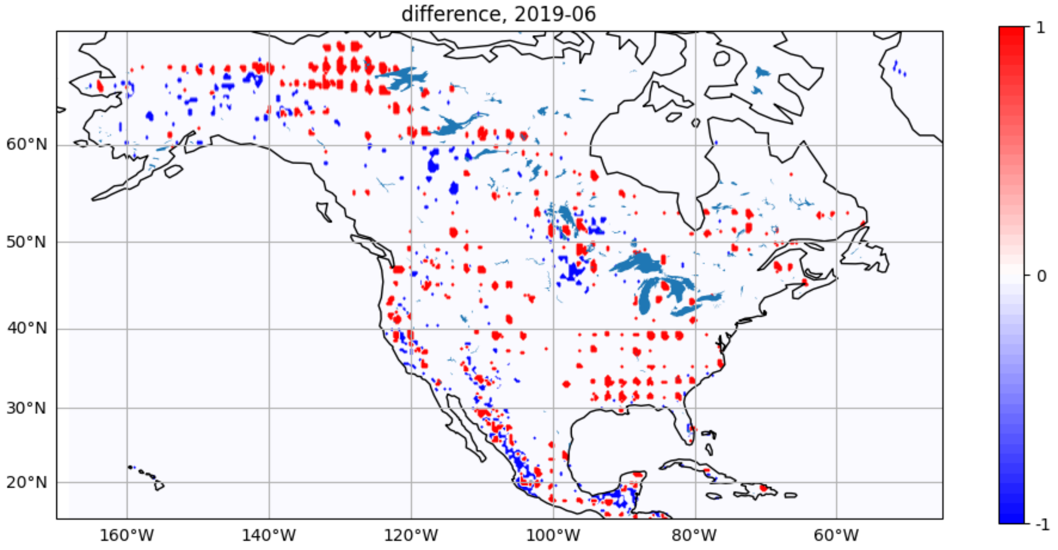
- Our initial try was to predict which pixels in an 8x8 tile were burning using a CNN
- We realised that this kind of generative CNN may not be optimal



−



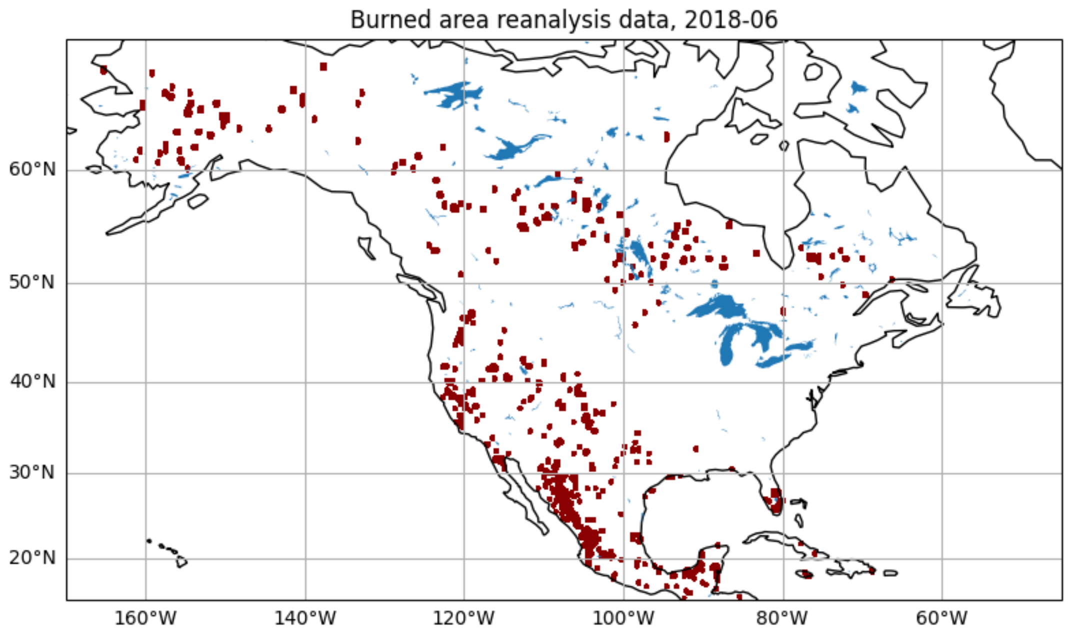
=



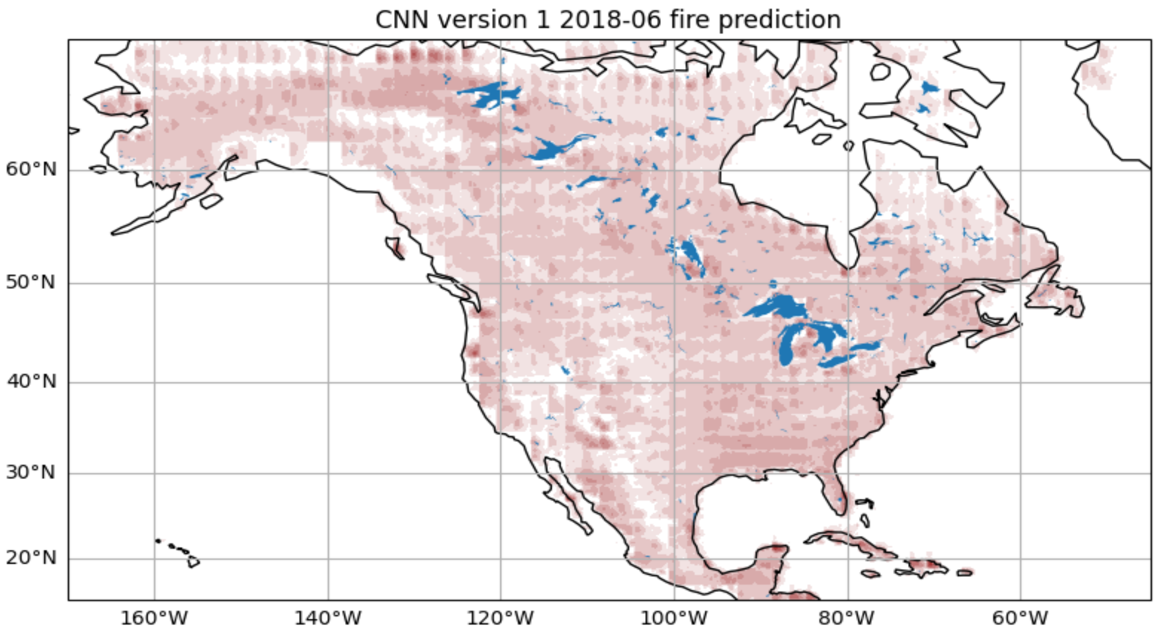
Subtract

Difference between true and predicted (0.29)

THE SPATIAL MODEL – CNN Version 1



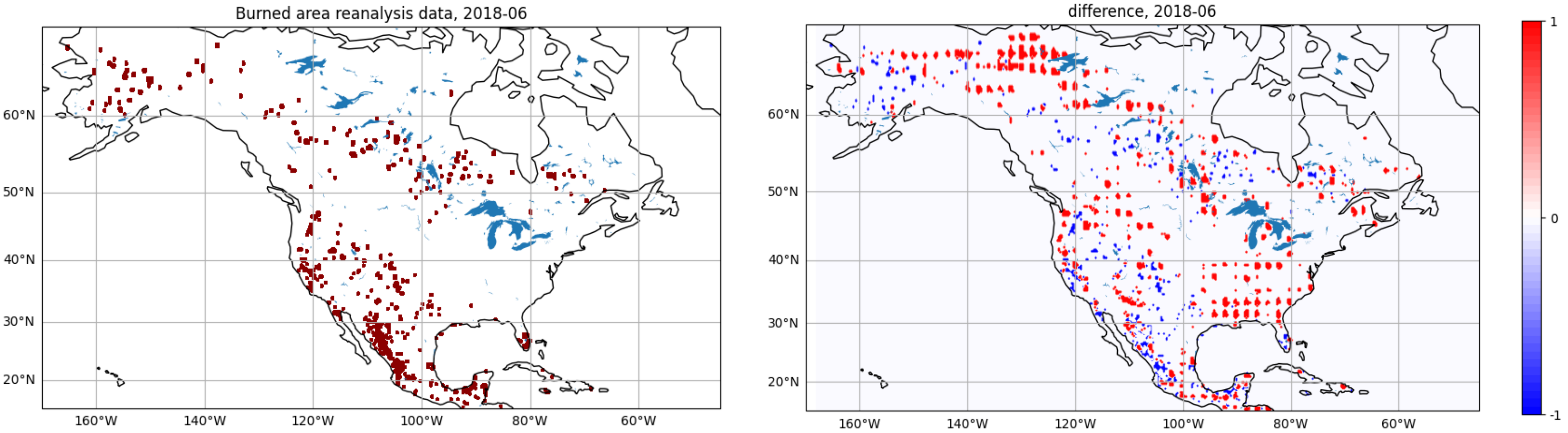
True Fire



Predicted Fire

› Same as in slides but for 2018

THE SPATIAL MODEL – CNN Version 1



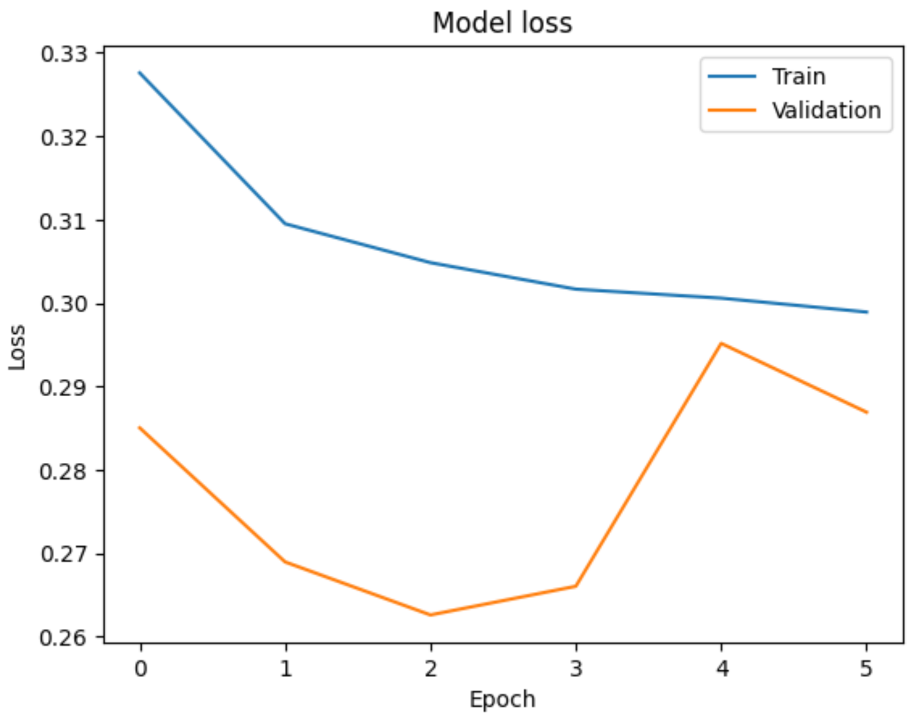
True Fire

Difference between true and predicted (0.29)

› Same as in slides but for 2018

THE SPATIAL MODEL – CNN Version 2

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 3, 3, 4)           940

 dropout_4 (Dropout)         (None, 3, 3, 4)           0

 conv2d_5 (Conv2D)           (None, 3, 3, 16)          592

 dropout_5 (Dropout)         (None, 3, 3, 16)          0

 conv2d_6 (Conv2D)           (None, 3, 3, 32)          4640

 conv2d_7 (Conv2D)           (None, 3, 3, 64)          18496

 dropout_6 (Dropout)         (None, 3, 3, 64)          0

 flatten_1 (Flatten)         (None, 576)               0

 dense_2 (Dense)             (None, 64)                36928

 dropout_7 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 61661 (240.86 KB)
Trainable params: 61661 (240.86 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



- Model Structure and Training and Validation Loss

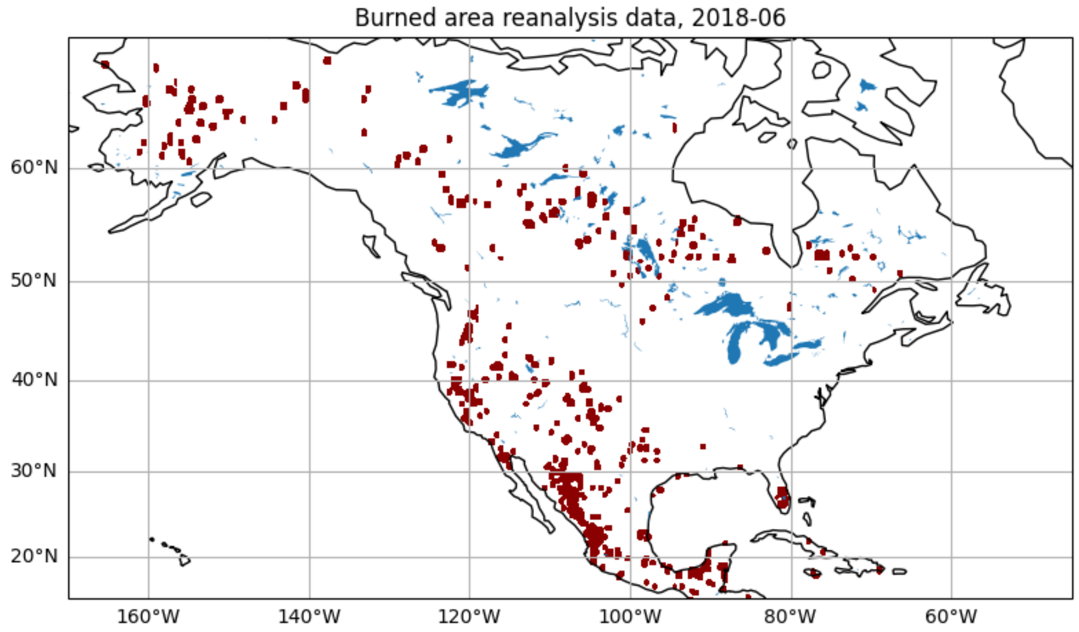
THE SPATIAL MODEL – CNN Version 2

- Target Balance: 3 Million No Fire : 200 000 Fire
- No Fire randomly selected from no Fire observations (> 10 Mio.)
- Train – Validation Split: 30%
- Data Augmentation: horizontal & vertical flip
- Saving Best Model
- Early Stopping based on validation data
- Weigthing of Classes based on Binary Cross Entropy (1.0 to 1.5)
- Loss = Log-loss
- Adam Optimizer for Step Size

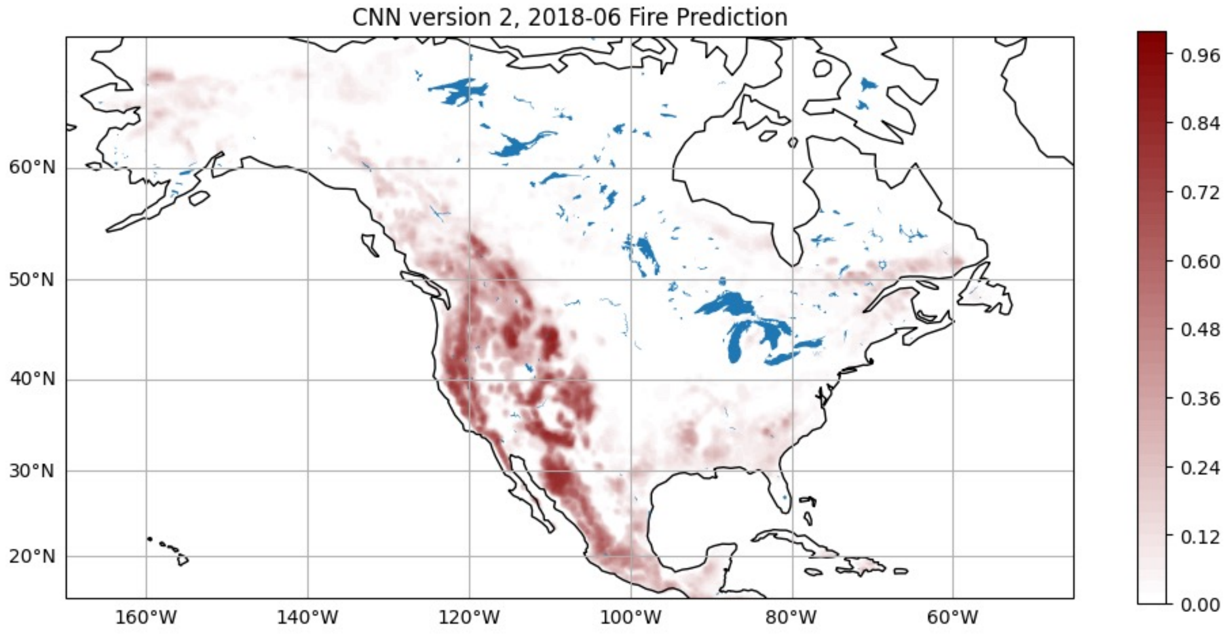
THE SPATIAL MODEL – CNN Version 2

```python
# ModelCheckpoint callback to save the best model based on validation loss
checkpoint = ModelCheckpoint('best_model.h5', monitor='val_loss', save_best_only=True, mode='min')

# EarlyStopping callback to stop training when validation loss stops improving
early_stopping = EarlyStopping(monitor='val_loss', patience=3, mode='min', restore_best_weights=True)


class_weight = {0: 1.0, 1: 1.5}  # Weigths of classes

history = model.fit(
    datagen.flow(X_train, y_train, batch_size=32),
    validation_data=datagen.flow(X_val, y_val, batch_size=32),
    epochs=100, batch_size=32,
    class_weight=class_weight,
    callbacks=[checkpoint, early_stopping]
)
```

THE SPATIAL MODEL – CNN Version 2



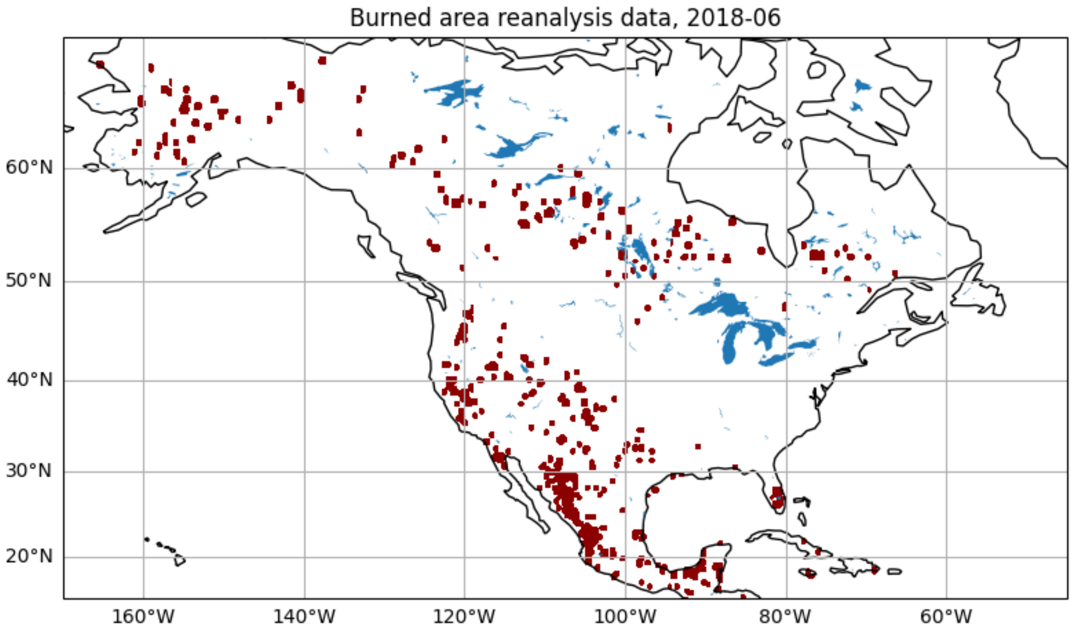
True Fire



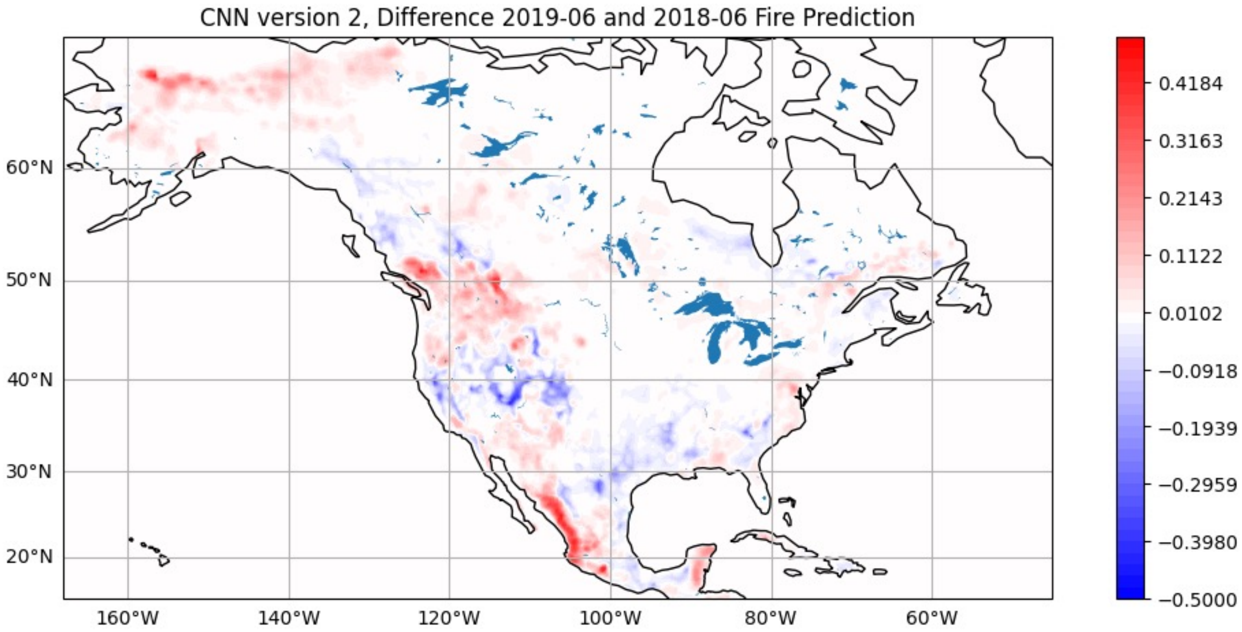
Predicted Fire

- Different Year

THE SPATIAL MODEL – CNN Version 2



True Fire



Difference in Predicted Fire

› Comparison between different years to make sure that different years predict different results.

CONTRIBUTION

- All group members contributed equally to the Project.

CODE

- https://github.com/Malus16/MLWildfirePrediction