# DS 4420 Project — Fish Doorbell

**Katheryn Lanman & Lon Pierson**
Northeastern University

## Introduction

From March to May, thousands of fish migrate through European waterways but are blocked by a lock in Utrecht, Netherlands. The city operates a "fish doorbell" — an underwater livestream camera that allows viewers to alert lock-keepers when fish need passage.
We developed a CNN to identify when fish are at the gate and a time-series analysis to predict peak migration times. By understanding migration patterns, we can improve viewer engagement with the doorbell and help ensure fish can safely continue their journey.

## Motivation and Data

**Why CNNs and Time-Series Models?**
CNNs are widely used for wildlife footage recognition. Prior work shows CNNs outperform other methods in detecting animal presence. Similarly, SARIMA models are the standard for predicting seasonal trends across many different fields.
**Data Sources:**

- **Live Stream Footage:** 4 hours of web scraped Utrecht Canal footage without fish + 30 minutes of fish-only footage.


**Figure 1:** Processed frame showing fish at the canal lock.

- **Migration:** StacomiR is an open-source project developed in France by the French Office for Biodiversity Institute to centralize data obtained by fish pass monitoring. We specifically pull data from the "r mig interannual vichy" dataset, which contains data from 1997 to 2012 on salmon migration at the Vichy counting station in the Loire River, France.

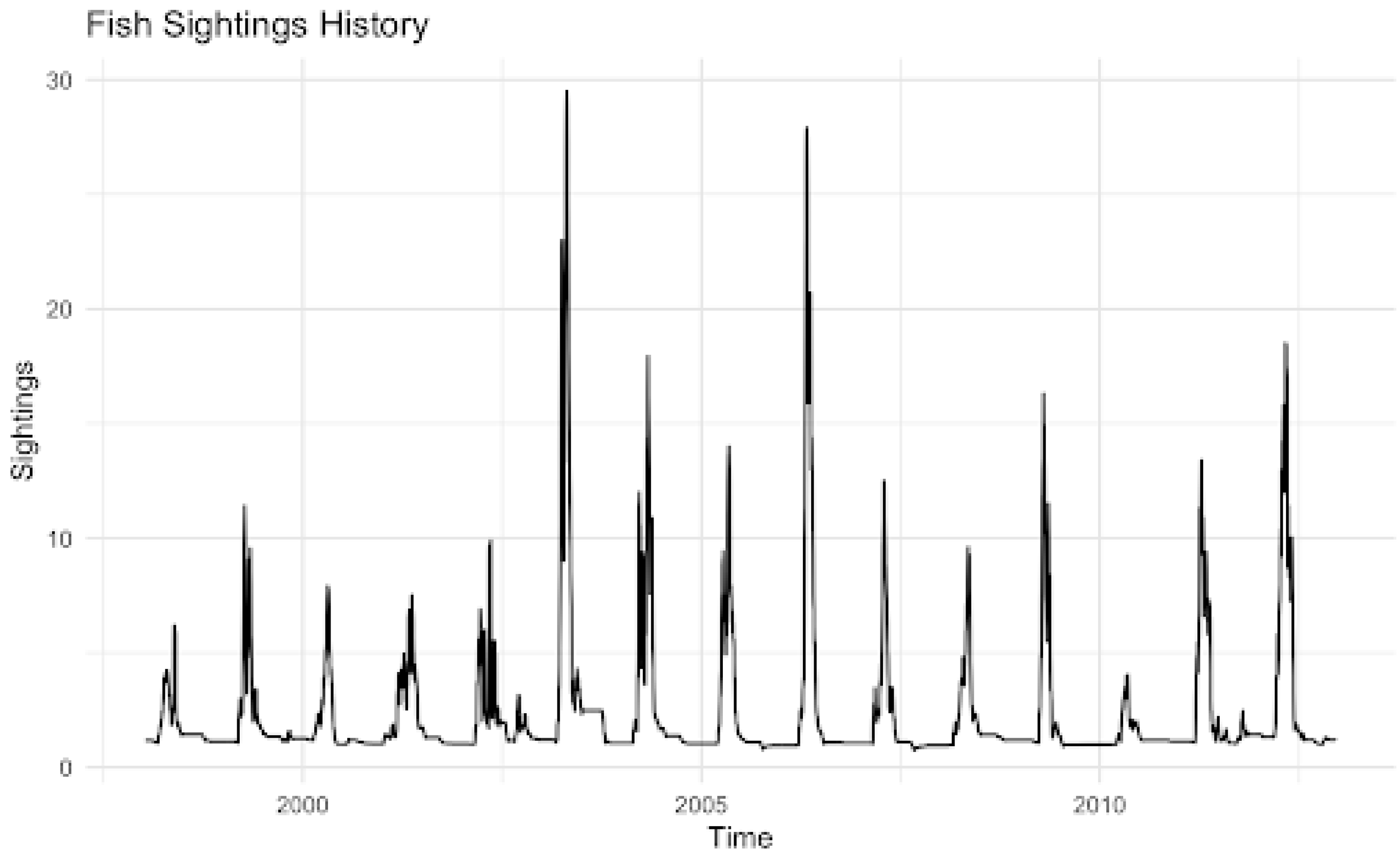


**Figure 2:** Fish sightings over time at the Vichy station.

## Methodology

**CNN for Fish Detection:** We created a few CNN models all with two convolutional layers to increase runtime. Each convolutional layer had a kernel size of 3x3 and a stride of one, with padding added to the first layer. Our models had 2–3 hidden layers, with the first layer containing 1000 nodes and the second having 500 nodes. If a third layer was added, it had 250 nodes. The models used a sigmoid activation function on the output since we're performing classification (fish or no fish).


**Figure 3:** Frame with a Fish present after 2 convolutional layers

**Time-Series Modeling:**

- **AR(365):** We first used this data to test an Autoregressive (AR) Model with a lag of 365 (since migratory seasons occur annually). To stabilize the wide variation in daily counts, we smoothed the data by calculating the 10-day moving average.
- **SARIMA:** Additionally, we tested a Seasonal Autoregressive Integrated Moving Average (SARIMA) Model with a seasonal differencing order of one, seasonal MA order of two, and seasonal AR order of two.

## Results

**CNN Performance:** After testing various models, we found that the best iteration was one with 3 hidden layers and the activation functions: sigmoid, ReLU, and ReLU, in that order. This model returned an average accuracy score of 1 and an average recall score of 1. We also manually inspected images to ensure correct classification. Figure 1 shows an example of a "fish frame" after two convolutional layers.
**AR(365) Model:** The AR(365) model produced a mean absolute error of 1.18 and a root mean squared error of 2.29. While the model seems to only miscount by one or two fish, the negative forward momentum shown in the graph below suggests the model may not be a good fit.

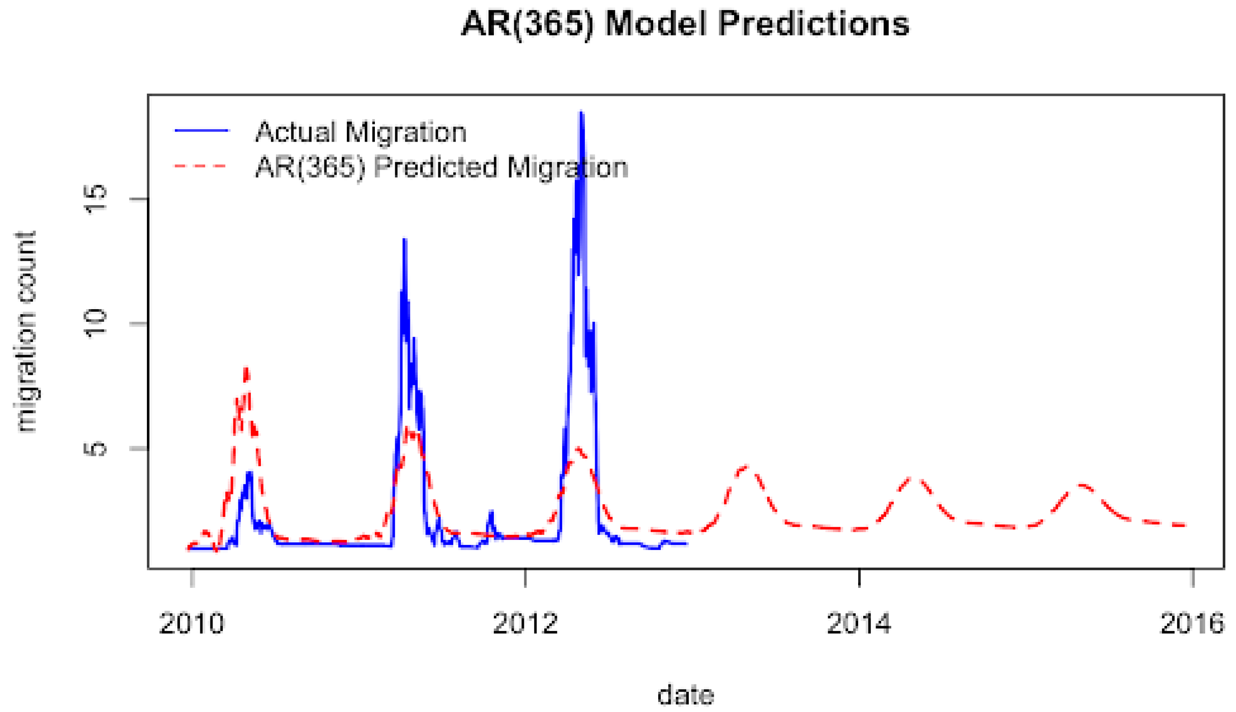


**Figure 4:** AR(365) predictions vs. true migration counts.

## Results

After fitting the data, the model had a mean absolute error of 7.08 and root mean squared error of 16.80. As shown below, the model correctly aligns with the seasonality of the data. Unlike the AR model, the predictions show fairly consistent trends across seasons with peak migration count staying steady. The SARIMA model also seems to be showing less dependency and better pattern recognition than the AR model.

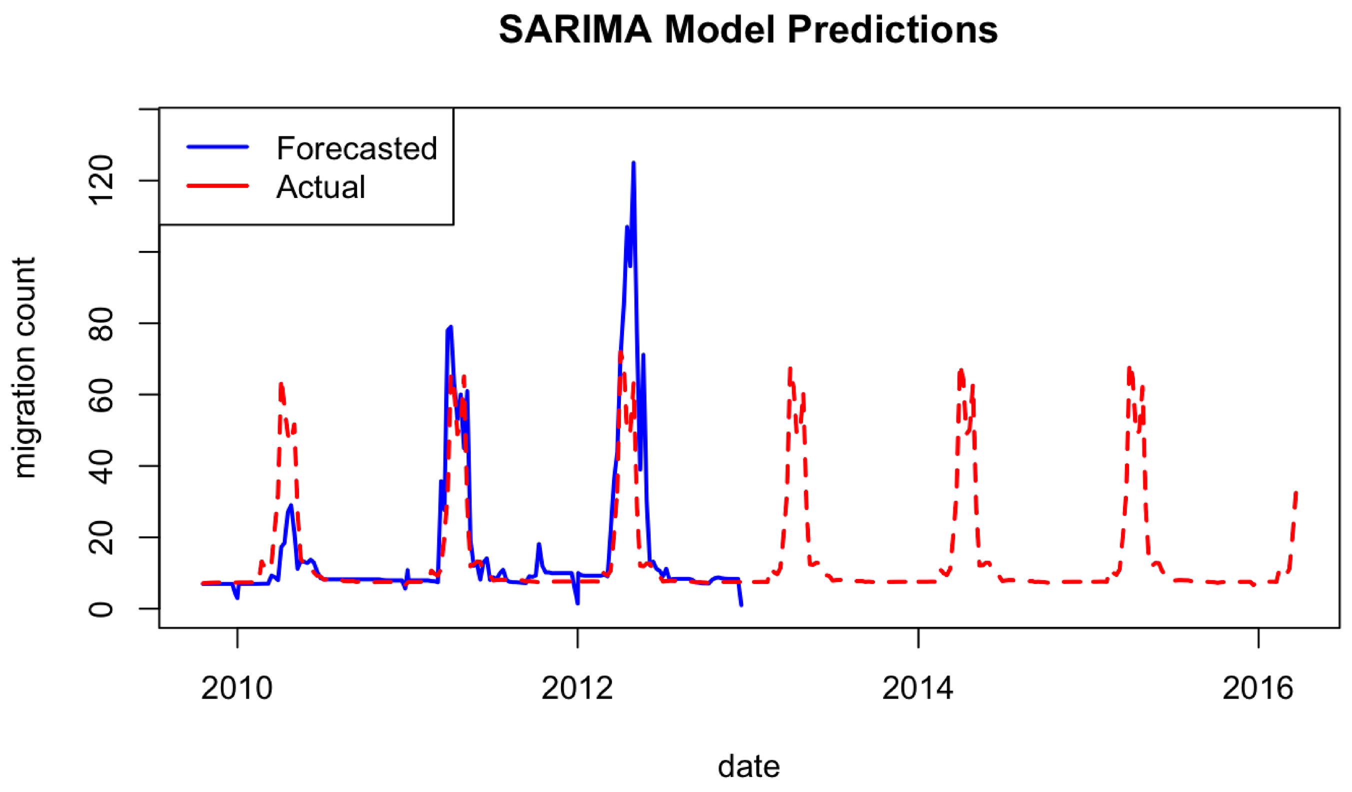


**Figure 5:** SARIMA model predictions with consistent seasonality.

## Discussion and Future Work

**CNN:** Future work on this would benefit from continuing to explore different model architectures to optimize runtime and memory usage. Another method of analysis we'd consider exploring in future work would be to take the difference in pixel values from one frame to the next to detect movement and train our model to detect movement instead.
**Timeseries:** Both the AR and SARIMA models were able to correctly fit seasonal patterns of the fish migration data. However, both models showed certain signs of poor fit, decreasing our confidence in their predictions. The AR model with a lag of 365, in particular, had trouble properly fitting the data, failing several of the assumptions necessary for producing confident results. If we were to continue working on this problem we would certainly look into a SARIMAX model to account for the variety in intensities of the migration spikes.
**Conclusion:** With additional data from the Utrecht Canal, we could develop a fully automated fish detection and prediction system, improving both engagement with the fish doorbell and helping fish migrate safely.

## References

1. Trancart, T., et al. (2013). Forecasting animal migration using SARIMAX. *Endang Species Res*, 21:181–190.
2. Trnovszky, T., et al. (2017). Animal recognition system based on CNN. *Adv. Electr. Eng.*, 15(3).
3. Xu, F., et al. (2021). Prediction of fish migration based on SARIMA model. *Complexity*, 2021(1).