horizontal line

**CSML1030: Machine Learning Capstone**

Sheep Transportation Tracking

Project Proposal for Canadian Sheep Federation (CSF)

Matin Mazid and Daniel (Jee) Lee

**10th April 2025**

This work aims to discover possible approaches to improving data processing accuracy by applying machine learning methodologies to handwritten forms. A step-by-step analysis will be conducted to identify patterns and inconsistencies, with a focus on ensuring accurate parsing and interpretation of the scanned form.

# PROJECT BACKGROUND

**Canadian Sheep Federation (CSF -** [**About CSF | cansheep**](https://www.cansheep.ca/about-csf)**)** helps farmers navigate the sheep industries, value chain. It uses handwritten forms submitted to CSF to track the movement of sheep. The forms contain large amounts of data, often encountering inconsistencies like how dates and time are formatted across different submissions. The lack of a standardized format can lead to errors in data extraction and processing.

For example, dates may appear in various formats, such as:

* **MM/DD/YYYY** (e.g., 03/10/2024)
* **DD-MM-YYYY** (e.g., 10-03-2024)
* **Month DD, YYYY** (e.g., March 10, 2024)

Similarly, times may appear in these formats:

* **24-hour format (Military Time):** Uses hours from 00 to 23, with no AM/PM.
* **12-hour format:** Uses hours from 1 to 12, with AM and PM.

These discrepancies pose problems when trying to standardize and extract accurate information from handwritten records. This creates a challenge for CSF, where misinterpreted dates could lead to inaccurate records.

In addition, the form allows submitters to circle values from amongst a set of values. This means we need to be able to identify selected circled text as the intended text.

# PROBLEM STATEMENT

Can-sheep’s requirement to track livestock through the value chain and keep robust and durable records has led to the use of paper records, which are scanned and then entered to the livestock tracing system. Paper records allow farmers and transportation professionals to track the movement of animals, without the technical problems of connectivity in rural areas. However this also means that CSF needs to manually enter transportation information into the CSF or automate their data input via Optical Character Recognition (OCR). Automated data input means that unstructured data like dates and text strings can be misinterpreted.

As a York University team, we are proposing an automated date format detection algorithm to help CSF overcome this issue. By analyzing differences between related date fields—such as departure and arrival dates—this approach enables dynamic and accurate date parsing without relying on rigid format assumptions. This solution aims to improve efficiency in processing diverse date formats.

The team can use image classification libraries to find circled text to provide values for questions that have multiple choices.

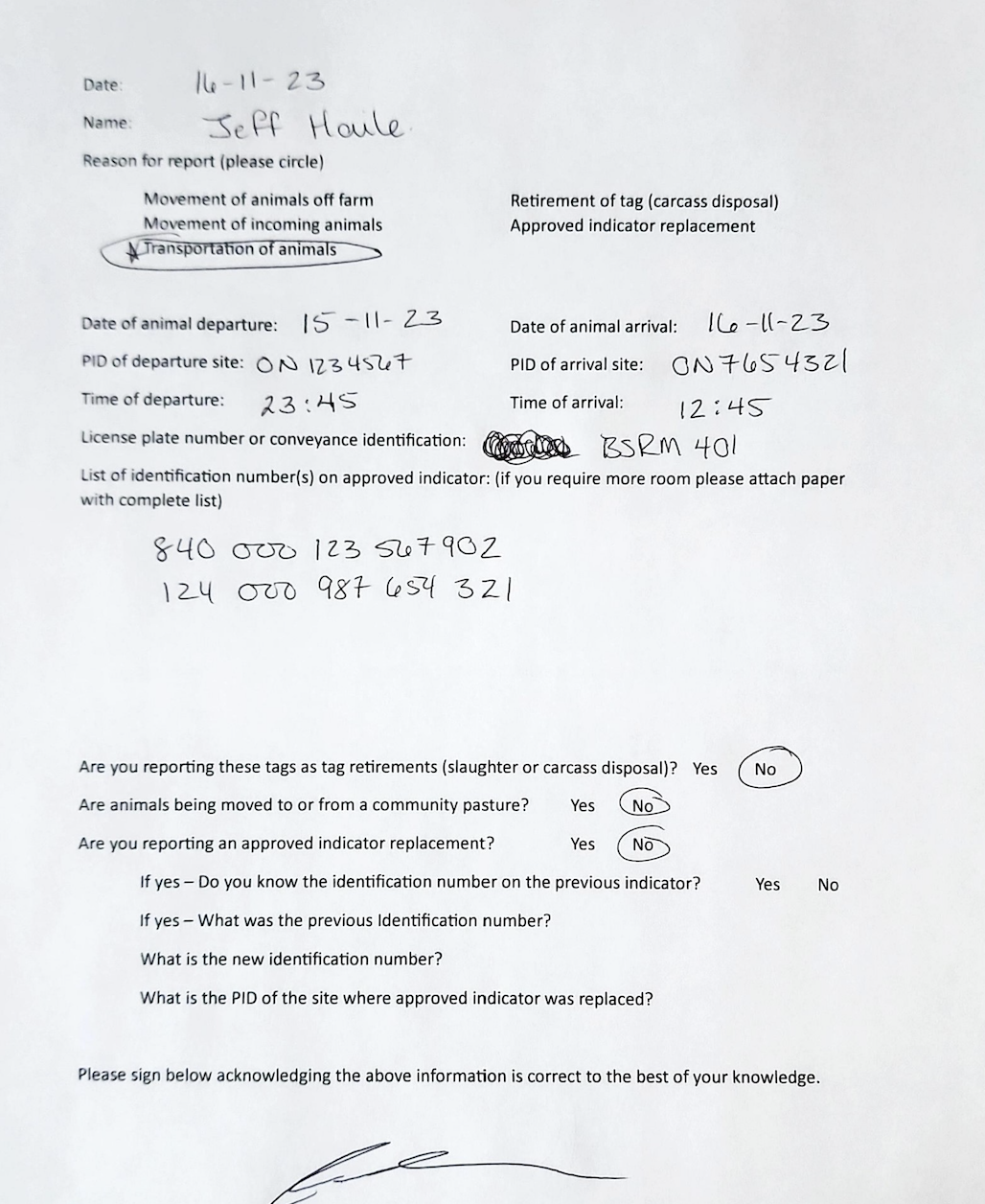
# SPECIAL CONSIDERATION

There will always be handwriting variability, as individuals have different writing styles and the quality of handwriting can vary greatly.

* **Legibility Issues**: Handwritten forms can be difficult to read, with varying handwriting styles and quality of writing (e.g., smudging, inconsistent characters). The algorithm might need to handle these variations effectively.
* **Character Recognition Errors**: OCR tools may misinterpret characters or digits (e.g., "1" vs "I" or "0" vs "O"), which can lead to errors in date parsing. Incorporating error-checking mechanisms could help improve the system.

# DATA DESCRIPTION

Image scans of the forms signed by a farmer are inputted as the data. A sample form is shown below



# GOALS

1. Populate and store the information found in the form.
2. Provide a heuristic to guess the date format so we can increase the operational accuracy of the information in the current CSF data store.
3. Start with a working back-end script before creating the final application that extracts handwritten data and stores it in the database.

# METHODOLOGY

## Methodology Highlights

* This approach enables robust detection of date formats across different input variations while ensuring chronological consistency.
* Its recommended that we use existing libraries to extract circled values from the main form
* The use of a human in the loop can spot interpretation errors and improve overall data quality

## Step 1: Extracting Dates from Image

Extract date from the long text read using OCR. The date fields can be identified from the surrounding text because the surrounding text will have a fixed content and so can be separated from the date. Dates that contain alphanumeric characters and easily be disambiguated. Dates that contain only numeric characters can be extracted and interpreted as described below.

## Step 2: Providing a process to identify dates

### 2a. Handling Alphanumeric Dates

If the date contains alphabetic characters (e.g., "Mar 10, 2024"), we extract structured components and focus on the numeric parts of the date.

### 2b. Provide heuristic to extract a parsable date

The algorithm assumes a consistent date format across all date fields. Departure and arrival dates are tokenized into their respective parts.

If we assume the transit duration is within 15 days, we can eliminate ambiguous date pairs. If we compare the maximum and minimum days between the two interpretations, we can eliminate any dates longer than 15 days. For example, if a form has the departure and arrival date as 10-03-24 (Oct 3 or March 10) and 11-03-24 (March 11 or Nov 3) respectively, we could be narrowed down to a March 10 departure and March 11 arrival because the other date combinations would be more than 15 days.

Also, we can look at the number of date fields that change. There are certain conditions where some or all the date fields change. For instance, assuming there is no more than 15 days of travel, the 3 fields of a date only change across 15 days of new years. Subtracting the arrival date from the departure date will only yield negative values when we cross into a new month and the month will only increment by 1 value within the same year. Therefore, we can find the day, month and year as long as the above assumptions hold.

**Identifying Name**

We may extract the name written on the form and use matching techniques to identify the most likely intended name. This allows us to correct common misinterpretation or formatting issues and match the input to known names in our database.

## Identifying Circled Text

We may extract the circled texts using existing libraries. The text can be mapped to a specific question.

Special consideration should be paid to the later questions because their “yes” , “no” response means we need to consider the order in order to map the response to the question.

## Identifying Times

The time fields and be populated by assuming either an AM or PM marker (eg. A or fully AM) or a 24 hour clock.

## Human In the Loop

Also, the application should allow staff members to review interpreted image data. This would allow us to correct poorly interpreted data.

# SUMMARY

Interpreting handwriting values will be a challenge and may demand a second set of human eyes to disambiguate or correct form data, however the pre described approach of disambiguating dates using assumptions based on business requirements of reasonably short travel time allows us to increase the accuracy of our date values.

# REFERENCE

[1] **Rosebrock, A.** (2018, September 17). *OpenCV OCR and text recognition with Tesseract*. PyImageSearch.<https://pyimagesearch.com/2018/09/17/opencv-ocr-and-text-recognition-with-tesseract/>