Machine vision in hay bale production

Benjamin Vail

Department of Biological and Agricultural Engineering, Kansas State University, Manhattan, Kansas, USA.

A paper from the Proceedings of the
16th International Conference on Precision Agriculture
21-24 July 2024
Manhattan, Kansas, United States

Abstract.

The goal of this project is to develop a system capable of real-time detection, pass/fail classification, and location tracking of large square hay bales under field conditions. First, a review of past and current methods of object detection was conducted. This led to the selection of the YOLO family of detectors for this project. The image dataset was collected through help from our sponsor, collection of images from the K-STATE research farm, and images collected from the internet. Training of the YOLOv8 model was conducted on a multi-GPU node of the K-STATE Beocat supercomputer.

For the prototype, the system would need to be small but still powerful enough to run real-time detection. The Nvidia Jetson Nano was selected to be the main computing platform for the system. The Luxonis Oak-D pro POE camera was chosen to carry out the detection and provide depth estimation of the bales for geotagging. With IR vision, IR laser active stereo vision, and an IP-67 rating, it is an excellent choice for use on agricultural equipment that might be exposed to the elements and can also support night-time baling operations. Results from testing showed that the system can detect good and bad bales, as well as provide GPS coordinates for each bale. However, changes in bale material such as corn stover vs alfalfa and the reduction of the model from the v8x to the v8n, impacted the performance of the system. The system was able to achieve a precision of 0.655 and recall of 0.799 during testing on corn stover bales.

For future work on this project, there are several avenues of improvement that would be advantageous. Design and testing of the system for low light/nighttime conditions. Upgrading the computer from the Jetson Nano to the Orin Nx would allow the model to be switched to the more accurate v8x and at a higher frame rate. And further work and fine-tuning of the image dataset.

Keywords.
Computer vision, Object detection, Forages, Baling, Hay, YOLOv8,

Introduction

Motivation:

The world's population is expected to reach 9.8 billion by 2050 and 10 billion by the end of the century (United Nations D. o., 2017). Even with the many advances in farming practices that have been made in recent history, we are still not currently able to produce enough food to feed that number of people. Meat, eggs, and dairy make up 14.5% of globally consumed calories on a yearly basis (United Nations F. a., 2021). In 2022, 18.5% of the $452.7 billion in annual farm production expenditure was due to feed costs associated with animal-based agriculture (NASS., 2023). In addition, roughly half of all livestock feed comes from harvested forages (Decision Innovation Solutions, 2020). This, coupled with the rising annual expenditure required of farmers since 2018 shown in Figure 1, means that efficient, cost effective, production and use of livestock feed resources is critical to feed the world in future generations.

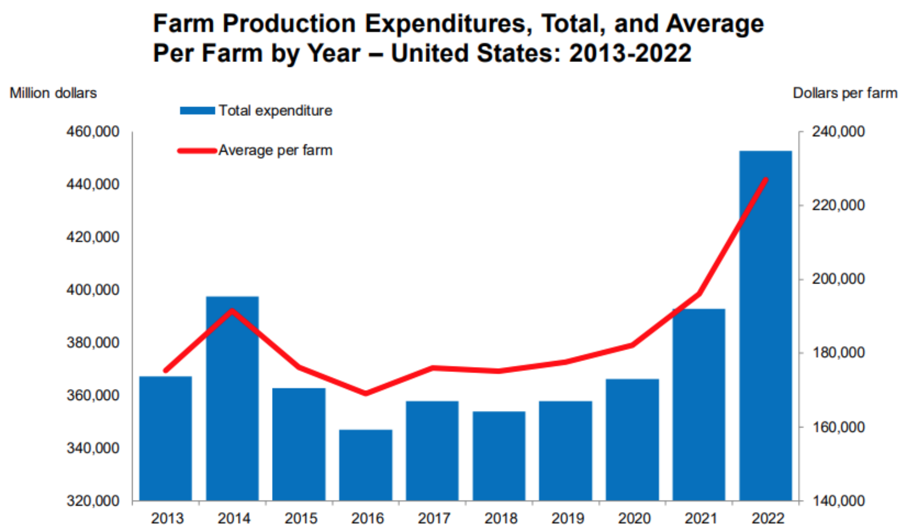

Figure 1: Farm Production Expenditures, Total, and Average Per Farm by Year – United States: 2013-2022 (NASS., 2023).

To meet increasing demands for food production while keeping the cost to farmers down, the efficiency of farm operations in general, and feed production in particular, will need to be greatly increased. To accomplish this, inputs such as labor, pesticides, fertilizer, and equipment costs will need to be minimized while outputs are increased. One of the current focuses of the agricultural engineering industry is smart/precision agriculture, or the use of AI and machine learning in farming. Technologies such as targeted pesticide spraying, automated equipment operation, and computer-vision aided disease detection are just a few examples that have been showing up on the market recently. Smart/precision agriculture and automation are proving to be effective methods of increasing productivity while lowering input costs, such as pesticides, fertilizer, and labor (Shin, et al., 2023).

Because feed is the number one expenditure in farming, these new smart/precision agricultural techniques need to be translated to the feed production industry to meet the increasing demand. One common method of handling harvested forages for feed is to form it into either round or square bales. This increases the density of the forage into a size and volume that is easier to transport and distribute. During bale production, incorrect settings, machine failure, or operator error can lead to bale failure. Bad bales, due to their irregular shape and density are harder to transport and often spoil faster than correctly baled forage. For this project, the focus will be on detecting any failures that cause a notable change in the bale's shape, such as twine failure, or major compaction errors.

Due to a request from our corporate sponsor AGCO Corp. and the unique challenges that square balers face over round bales, we will be focusing on square hay bales for this research. Machine settings such as bale compaction, twine tension, and feed rate, are a particular concern for square hay bales. These failures necessitate either manually completing the process, or in the worst case, the breaking down of the bale and then attempting to re-bale the forage. Both cases increase production costs as well leading to increased loss of forage, but re-baling is particularly costly. And if the failure is a recurring issue and the operator doesn't detect the problem quickly, a large number of failed bales can potentially be produced. The costs associated with these failures have created a need for research into ways to quickly identify bale failure during the production process and alert the baling operator.

Objectives:

While there are mechanical methods capable of detecting bale failure, this only works while the bale is in contact with the system. Many bale failures can occur after the bale has left the hay baler and so a system able to continue monitoring for an amount of time after dropping the bale is necessary. To quickly detect bale failures and alert the operator, a detection system would need to be able to recognize what parameters determine a good hay bale and be able to detect any deviation from those parameters. This system, to reduce distractions faced by the operator during baling, should also require as little interaction by the operator as possible. For this application, computer vision is uniquely suited, being able to continue monitoring the bale as long as the bale continues to be in the camera's field of view and within a reasonable distance.

Computer vision has been a growing area of research in agriculture for many years now, and technology is rapidly advancing. In the preliminary stages, it was heavily impacted by lighting, movement, camera obstruction, high hardware requirements and changing viewing angles. These shortcomings of older computer-vision methods often prevented the use of computer-vision systems under field conditions, but advances in the past decade and even just the past few years have negated many of these problems. Newer technologies, such as deep learning architectures, improved computing power and miniaturization of integrated computing platforms, enable computer vision to be utilized under field conditions.

The goals for this project are as follows:

- Research common and emerging computer vision methods available to the market and select the best option for use in the detection of hay bales and performing pass/fail analysis of the finished bale. To perform this pass/fail analysis, the system should monitor the bale's shape for as long as possible.
- Collect an image dataset of both good and failed hay bales for the training of the previously selected computer vision model with a high degree of accuracy, precision, and recall metrics.
- Design and implement a prototype hardware system capable of deploying the trained model under field conditions. The prototype system must be capable of withstanding field environmental conditions, such as dust, water, and vibration. The system must also be able to determine the current GPS coordinates of the detected hay bale and record those coordinates along with the pass/fail determination of that bale for later retrieval.

Background:

Bale Failure Causes

Square hay baling operation begins by picking up the loose forage windrow with feeding tines, which is then carried into the bale chamber. In the bale chamber, a ram arm compresses the newly added forage material into the end of the bale. While the bale is being compressed, twine is fed through the chamber to the knotter and at an operator specified tension. Once the end of the current bale is reached, the twine is cut and knotted off to secure the bale. The completed bale is then ejected out of the back of the machine.

A bale can be considered a failure for several reasons; if the shape is not square and uniform, if the bale is larger or smaller than the desired dimensions, or if the bale falls apart upon ejection from the system. Each step of the baling process can experience failures caused by incorrect settings, system failure, or operator errors, such as:

1. During the forage pickup, moving at speeds greater than suggested rates can cause some of the forage to not be picked up properly. This type of failure results in crop loss rather than bale failure and will not be addressed in this project.
2. Inconsistent feeding rates into the chamber due to uneven windrow volume can cause changes in layer compression of the bale. This can lead to bales that are too tightly compressed and may break the twine, which is a contributing factor to failed bales.
3. Forage moisture levels necessitate changes to the feed rate into the bale chamber. If these are incorrectly adjusted, bale failure can occur, either during the baling process or afterwards when the forage moisture level changes.
4. Bale chamber tension can be improperly set by the operator, which can lead to incorrectly compressed bales. This can lead to mis-shaped bales, or broken twine failures due to over compression.
5. The twine tensioner being incorrectly set can lead to broken twine failures.
6. Wear or damage to the twine knotters can lead to a failure in securing of the twine at the end of the baling process which will cause the bale to fall apart once ejected.

Bale Failure Costs

Estimating bale failure costs can be difficult due to the varied nature of the forage type, moisture content variability, skill in handling of the operator, and machine dry matter loss rates. In late 2023, Alfalfa prices for large square bales was between $200 and $325 per ton dry matter (USDA, 2023). If the operator has someone standing by that can unbind the failed bale, loosen up the forage matter, and place it onto the windrow, then it can be re-baled without needing to stop the baling process. This means that at minimum, the failure has resulted in doubling the baling twine and machine run time costs of that forage material while also requiring an assistant to process the broken bale. This doesn't consider dry matter loss during the re-baling process, where some loss is inevitable. Using an average of 2 tons or 4 large square bales per acre, 10 minute of work time by a farmhand to breakdown a bale, and 108ft of baling twine (3'x4'x6' square bale), the cost per fixing a bad bale is $11 (see Table 1)

	Cost	*Source*
Farm hand Pay (est. 10 min./bad bale)	$2.28	(Payscale, 2023)
Baling Twine (3x4x6 6 twine square)	$2.07	(CWC, 2023)
Machine (est. 3 min./bale, 2 bales/acre)	$6.65	(NCSU, 2013)
Total	$11.00	

Table 1: Bale Failure Cost Analysis.

Deep Learning Methods:

Deep learning methods, since their introduction in 2012, have quickly taken over as the focus of research and implementation in computer vision. Their ability to automatically learn features from a dataset without being dependent on user input makes them far more versatile than most shallow learning methods. Deep learning architectures can take in multiple feature domains, and through multi-layer abstraction, create complex feature hierarchies (Picon, Alvarez-Gila, Irusta, &

Echazarra, 2020). Their shallow learning counterparts were often limited to just one or two domains, and if more were needed, then often multiple models would have to be run simultaneously, and their inputs combined (Picon, Alvarez-Gila, Irusta, & Echazarra, 2020).

In the current market, there are many different deep learning architectures and even more custom variants of each. Out of this plethora of architectures, there are a few that have received more focus from the industry and research community than others, such as the two-stage RCNN and single-stage YOLO architecture families. To reduce costs to the end user, the detection algorithm should be as open-source as possible, and it needs to be accurate while still running in real-time. The task of detecting hay bales also requires consideration for the environment that the equipment will be operating in. Such as needing whole-image context during training, since hay bales are similar to the background of the field they are produced from. With these guidelines in mind, it was determined that the YOLO architecture family; with its simplified detection pipeline, short latency, and relatively high degree of accuracy, would be a good match.

YOLO Architecture

The YOLO architecture was first introduced by Joseph Redmon et.al. in their 2016 paper "You Only Look Once: Unified, Real-Time Object Detection". Most other architectures of the time were using one network to propose regions that are most likely to contain the object and then a second to determine the actual location and bounding region. Redmon and his team, instead proposed treating detection as a regression problem and conducting the process with a single pipeline using one network. YOLO does this by splitting the image into an SxS grid and determining which cell of the grid contains the center of the object, and then moves out from the center, determining the bounding region of the object.

YOLO has undergone many improvements since its inception, with the most current version being YOLOv8 (Jocher, Chaurasia, & Qiu, 2023). At the start of this project YOLOv5 was the current version and was used to train a model on a dataset of images gathered from the K-State dairy farm, online image repositories, and images supplied by AGCO corp. In January 2023, Ultralytics released YOLOv8, which showed a significant improvement in detection speed over previous versions (Jocher, Chaurasia, & Qiu, 2023) (see Figure 2). Due to the limited size of the hardware being used, it was decided that the increased speed but comparable size and hardware requirements of YOLOv8 was worth updating the model.

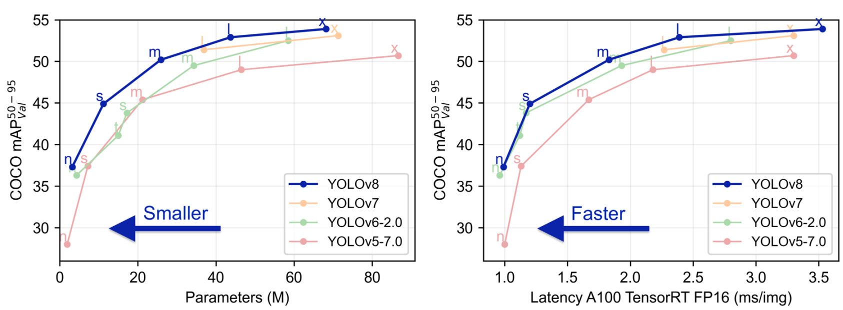

Figure 2: YOLOv8 latency performance vs. previous YOLO versions (Jocher, Chaurasia, & Qiu, 2023)

YOLO Backbone

The backbone of the YOLOv8 detection architecture is composed of P1 through P5 sections with 168 total layers, 53 of which are convolutional layers (King, 2023). P1 and P2 are base convolutional layers with the neck and head layers P3 through P5 interlinked to form a combined header layer (King, 2023).

The diagram in Figure 3, by RangeKing of GitHub in cooperation with one of the YOLOv8 developers Glenn Jocher, shows the composition of the YOLOv8 structure (King, 2023). The first two layers extract object features from the image. The SPPF layer and its convolutional layers process those features at multiple scales. The up-sample layers process the feature maps at increasing resolutions. The C2F layers combine features with contextual information gathered from the image background to improve accuracy. Once all of this has been done, the detection layers then map the features to a bounding box and object class.

Training of a model involves running this structure over a large dataset of images that have been annotated to include a bounding box around the object and the object class name. The structure extracts object features from the bounding box area while also gathering contextual information from outside the bounding box and then condenses this into a detection model. Contextual information can be especially important to the accuracy of a model, which is why many groups will provide pre-training on other image datasets that don't include the object being looked for but have similar environments or backgrounds. This pre-training gives the model better context for the environment in the background and what isn't the object being searched for compared to training from scratch. This model is then run over a separate validation dataset for accuracy checks. This process is run in segments known as epochs, where one epoch is when the algorithm has trained once on every image in the training dataset. YOLOv8 continues training until the number of epochs set by the user or until there hasn't been an increase in mAP50, precision, or recall for a defined number of epochs.

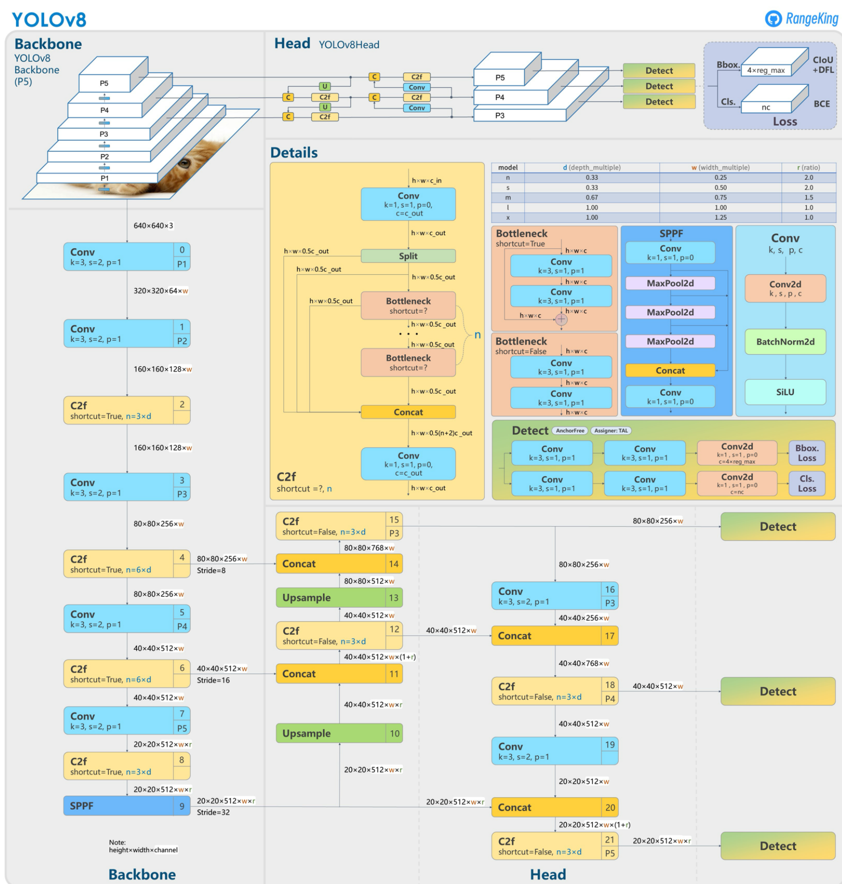

Figure 3: YOLOv8 Backbone Structure (King, 2023).

Training Results:

An image dataset of over 6000 hay bale images was assembled from images supplied by AGCO Corp., images taken of hay bales at the K-State research farms, and images collected from the internet. To provide a good baseline for further testing, two separate baselines were trained; one that was trained on the data without pretraining on a commonly used dataset such as MSCOCO, and one that did include pretraining with MSCOCO. Because the dataset that was collected was from such a limited number of sources, the overall training statistics showed an unusually high-level of accuracy, precision, and mAP50, even on images from the set that had been set aside for testing. To run a proper test of the system, a set of test images was obtained from the internet. This test set was comprised of images that were from different fields and baler manufacturers than what was used in the training dataset. This new set was then used so that the test dataset would be as unrelated to the training dataset as possible. These tests showed a drop in mAP50 in comparison to the original test dataset statistics as expected, providing a more realistic test of the models (see Table 2).

Class	*Precision*	*Recall*	*mAP50*
All	**0.858**	**0.629**	**0.762**
Good Bale	**0.737**	**0.601**	**0.671**
Bad Bale	**0.979**	**0.657**	**0.853**

Table 2: YOLOv8 model metrics.

Field-Test Results:

Once the design and construction of the prototype was complete, testing was performed with support from AGCO Corp. The initial prototype was mounted to one of their large square balers and run through multiple baling operations in the Great Bend Kansas area. The testing was delayed multiple times due to rainy conditions, but the fields had completely dried out by the time baling commenced. Overall field conditions were sunny and dry during the baling, with the soil being dry enough to see minor dust kick-up during baling. The forage being baled during this process was corn stover, while the model had mostly been trained on alfalfa. This allowed testing of the model's ability to generalize hay bales and how different forage species will impact detection. The system saves an image and the associated bale's GPS coordinates every 4 seconds while an object is detected in the ROI area of the screen. Over the baling process, 642 images were collected, although the last 13 images had to be discarded due to the camera being displaced during over-night storage and were unusable. Over the trial, the detection of good bales showed impressive precision but a lowered recall due to a larger than normal number of false negatives. These false negatives were due to the system over-classifying the corn stover bales as bad, leading to increased bad bale false positives and good bale false negatives.

Figure 4: Example detection image from the AGCO test of the prototype system.

	Precision	*Recall*
All	0.655	0.799
Good Bale	0.899	0.683
Bad Bale	0.410	0.914

Table 3: Prototype testing statistics.

Conclusion:

While the resulting precision and recall metrics were lower than the original training validation statistics, that was to be expected with the bales being corn stover rather than alfalfa and on the smaller YOLOv8n model rather than the YOLOv8x. A better comparison is to the test statistics run on images collected from random internet hay bale images (see Table 2). The results from that test showed (over-all precision = 0.858, recall = 0.629). A lower precision and higher recall compared to the values in Table 3, shows that it was having a larger number of false positives but a lower number of false negatives. Meaning that it was misclassifying the corn stover bales more often but failing to detect bales less often.

The driving force behind these results can be broken down to two main reasons; the material of the bales (corn stover) being different from the material of the training bales (alfalfa), and the much smaller number of parameters of the YOLOv8n model compared to the YOLOv8x model that was used in the training and augmentation phases. The YOLOv8n model has 3.2 million parameters while the YOLOv8x model has 68.2 million parameters. This smaller model was chosen so that it would be able to run on the Luxonis Oak-D pro camera that was used in the prototype, but this will need to be re-evaluated moving forward.

Dust generated during field operations obscuring the camera's view was a concern that we wanted to investigate but did not have a way of testing until field testing. During prototype testing, several images were collected with dust being blown between the bale and camera. This dust was not thick enough to obscure the bale from the camera.

References

CWC. (2023). *Big Baler Twine*. Retrieved from CWCGlobal: https://www.cwcglobal.com/twine/baling/big-baler

Decision Innovation Solutions. (2020). *Animal Feed/Food Consumption and COVIS-19 Impact Analysis.* Retrieved April 4, 2023, from Decision-Innovation: http://www.decision-innovation.com/webres/File/docs/210525%20IFEEDER%20Animal%20Feed-Food%20Consumption%20COVID-19.pdf

Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO. (8.0.0). Retrieved from https://github.com/ultralytics/ultralytics

King, R. (2023). *Brief summary of YOLOv8 model structure*. Retrieved July 2023, from https://github.com/ultralytics/ultralytics/issues/189

NASS. (2023). *Farm Production Expenditures 2022 Summary.* USDA., NASS. Retrieved from https://usda.library.cornell.edu/concern/publications/qz20ss48r

NCSU. (2013). *Hay Harvest Costs.* NCSU, Extension. Retrieved from https://cals.ncsu.edu/are-extension/wp-content/uploads/sites/27/2018/03/ForageBudHayLRB84-2_Print_2013_0.pdf

Payscale. (2023, October 25). *Average Farm Hand Hourly Pay*. Retrieved from Payscale: https://www.payscale.com/research/US/Job=Farm_Hand/Hourly_Rate

Picon, A., Alvarez-Gila, A., Irusta, U., & Echazarra, J. (2020). Why deep learning performs better than classical machine learning? *Dyna Ingenieria E Industria*. doi:http://dx.doi.org/10.6036/9574

Shin, J., Mahmud, M., Rehman, T., Ravichandran, P., Heung, B., & Chang, Y. (2023). Trends and Prospect of Machine Vision Technology for Stresses and Diseases Detection in Precision Agriculture. *AgriEngineering*(5), 20-39. doi:https://doi.org/10.3390/agriengineering5010003

United Nations, D. o. (2017). World Population Prospects: The 2017 Revision, Key Fidnings and Advance Tables. *United Nations*.

United Nations, F. a. (2021). Statistical Yearbook: World Food and Agriculture 2021. *FAO*.

USDA. (2023). *Kansas Direct Hay Report.* USDA, Ks Dept of Ag Market News. Manhattan: USDA. Retrieved November 2023, from https://www.ams.usda.gov/mnreports/ams_2885.pdf