Machine learning case studies

How companies leverage data to drive outstanding results.

tryo·labs

“Those who can imagine anything,
can create the impossible.”

Alan Turing

Index

Introduction

Using data to drive positive business results is discussed widely, and many companies have recognized the potential of artificial intelligence — and **machine learning** in particular — to implement **powerful solutions based on data**.

In practice, companies can benefit from a wide range of machine learning applications but it's not always obvious which solutions are the most feasible and effective for achieving the desired results. The **best techniques available to solve a problem vary** depending on the companies' individual goals, resources, and data.

To get an idea of how machine learning can create business value in different industries utilizing different techniques, we came up with a summary of some projects that we have carried out at Tryolabs. Each case study touches upon the company's challenge, the **custom machine learning system** we built, the results, and the approach we chose to achieve them.

All solutions follow the steps outlined below:

- Understanding the business challenge.
- Evaluating different approaches to solve the challenge.
- Building proofs of concepts or production-ready systems.
- Evaluating results and performing additional iterations, if necessary.

Some case studies consist of **end-to-end solutions** that we implemented from scratch and put into production, while others are **proofs of concept** used by companies to establish machine learning roadmaps.

We hope the following case studies help you become more familiar with the tremendous opportunities that exist around using data to drive business results and encourage you to jump in on a machine learning endeavor in your organization.

Case study 1

Setting retail prices

Solution built for a large online consignment marketplace, headquartered in San Francisco, CA.

Results

- Ability to price more than **90k items per month** without human supervision.
- Achieving consistency in pricing.

Approach

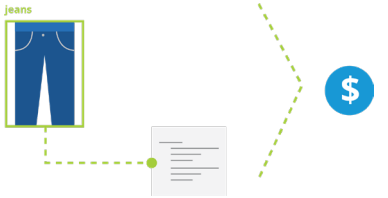
Create, train, and deploy a machine learning pipeline along with an entirely new infrastructure, capable of **determining prices for new items based on product images, product information, and historical sales data**.

Before

Prices of unique products in an extensive catalog are **manually determined** in an extremely time-consuming process. Pricing is neither cost-effective nor consistent.

After

More than half of the **products in the catalog are priced automatically**. Prices are determined based on product images, prices of visually similar products, product descriptions, and other attributes.



Technical details

Train convolutional neural networks to classify products and extract relevant features from images (e.g. pattern, neckline, sleeve length). Use image similarity and K-nearest neighbors (k-NN) to find the most visually similar products and their prices in the sales database in order to provide quality inputs for the algorithm.

Application of tree-based gradient boosting algorithms to predict prices based on the formerly retrieved data.

Case study 2

Building an automated category tree

Solution built for Mercado Libre, the largest online retailer in Latin America, headquartered in Buenos Aires.

mercado libre

Before

Inefficient system of **manual item category** creation in 16 very large category trees (one tree for each country the company operates in). Ineffective re-categorization of products and monotonous allocation of new items to categories.

After

Autonomous category creation as well as efficient product classification and proper categorization of millions of new or previously misclassified items with little to no human supervision.

Results

- **10x faster** creation of new item categories.
- Daily re-categorization of **25k new items**.
- **5x** increase in **productivity**.

Approach

Analyze massive amounts of historical data and implement a system to automatically suggest missing categories.

Create a **model using product titles, product descriptions and users' navigation data to improve manual categorization efficiency**.

Build an intuitive and intelligent front-end allowing for human validation of the automated category suggestions, so that the system can learn from feedback.

Technical details

Experiment with various algorithms in order to generate new categories: K-Means, hierarchical clustering, suffix tree clustering, custom vectorization techniques, and distance functions.

R&D finally leads to the creation of a custom overlapping clustering algorithm using neural networks and users' navigation data preprocessed directly from Hadoop.

Implement a recursive neural network (LSTM) for item classification, relying on page views and item data as inputs.

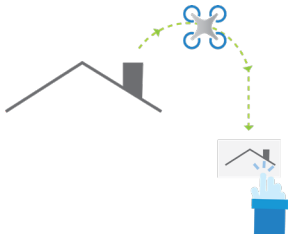
Case study 3

Scaling image processing used in roof inspections

Solution built for a risk management company, based in San Francisco, CA.

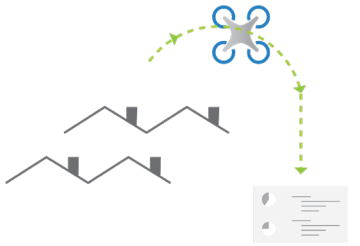
Before

Manual analysis of pictures of building roofs taken by drones to detect damages. Service offered by company is not scalable due to time consuming nature of analysis and having limited human resources.



After

System able to understand images and **automatically identify issues on roofs** including water pooling, loose cables, and rust.



Results

- Analysis of thousands of images simultaneously.
- Company is able to **scale services and increase order volume**.

Approach

Develop a **deep neural network capable of detecting several issues in roof imagery**, as well as a pipeline to analyze incoming images and an API to give external tools access to the results.

Technical details

Given the lack of a preexisting dataset, the first step involves creating a custom dataset. To facilitate the tagging process, the segmentation problem is approached as a classification problem by classifying small image patches instead of pixel by pixel (very time consuming).

Even though it's a much faster process, creating a big enough dataset to train a deep neural network from scratch is still very expensive. To improve the training process, a ResNet-18 model pre-trained on ImageNet is used. When applying the transfer learning technique, the model is trained to capture relevant information from a big dataset (ImageNet), then fine-tuned on a small dataset to learn specific information unique to the problem at hand.

The model mentioned above is employed as a feature extractor. Once good patch representations are generated, a simpler classifier is trained on them to generate the final result.

Case study 4

Predictive maintenance of oil well pumps

Solution built for Halliburton, an oil service company, headquartered in Houston, TX.

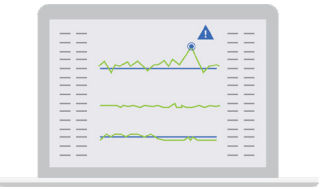
HALLIBURTON

Before

Data collected from sensors in oil well pumps is analyzed manually by employees in order to assess the health of the machinery. Ability to detect faults at scale is very limited.

After

Automated analysis of several dozen metrics collected from the pumps and sending of **notifications to operators in case of warning signs**. Engineers have access to a dashboard that displays health of machinery and problematic values.



Results

- Capacity to analyze data from **2000+ wells** in near real time.
- Ability to detect **new relevant alerts**.
- Reducing false positive alerts by 50%.

Approach

Build a custom front-end for petroleum engineers to tag or label regions of data of interest. Use data to train a machine learning **model that keeps track of changes in performance metrics** and predicts potentially unhealthy scenarios.

Technical details

Develop a feature extractor for time series data, using NumPy, SciPy, and Pandas.

Build a rules engine using time series analysis, as well as machine learning models, to automatically detect events and, from those events, generate different types of alerts related to the health of the wells.

Implement a React application to display current metrics for the different oil well pumps and the generated alerts, as well as to enable operators to give feedback on alerts.

The UI additionally allows operators to use their expert knowledge to tag healthy/unhealthy time ranges, along with other possible predefined points of interest. The labels provided by the operators are leveraged to implement an automatic feedback loop.

Retouching product images

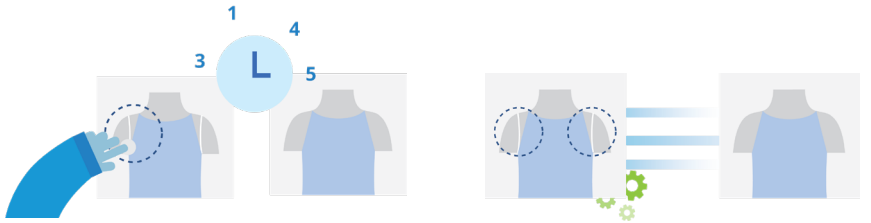
Solution built for a large online consignment marketplace, headquartered in San Francisco, CA.

Before

Pictures taken of mannequins wearing clothing items for sale are **retouched by an editing team** before they are uploaded to an online catalogue. High costs for third-party editing services.

After

Scalable **system with the ability to perform several retouching tasks**, such as background cleaning, image cropping or mannequin alignment, without human assistance.



Results

- **Eliminating expenses** for third-party editing services.
- Obtaining **thousands of retouched images** within seconds.

Approach

Create, train, and deploy a computer vision **system capable of retouching photographs of mannequins wearing product items**. Editing includes background cleaning, image cropping, mannequin alignment, removal of the base and markings of the mannequin's joints.

Technical details

Using the raw as well as manually edited photographs to train multiple neural networks:

- FCN and U-Net detect and segment parts in photographs that need to be edited, such as the base, the mannequin shoulder seams, and clothing.
- ResNet determines the appropriate cropping for each image.

Segmentation of shoulder seams turns out to be a challenge, since training data is not perfect and sometimes the network predictions don't cover the seam entirely. This challenge is overcome by expanding the mask in the direction of the shoulder seam, which is determined using PCA and fitting a line into the shoulder seam using Hough transform.

Additionally, classical image processing techniques such as adaptive gaussian thresholding and image filtering are applied to clean the background.

Keras is used to define the neural network architectures, while the data loading and training process is performed in Google Cloud.

Case study 6

Anticipating emergencies

Solution built for GreatCall, a health technology provider, headquartered in San Diego, CA.

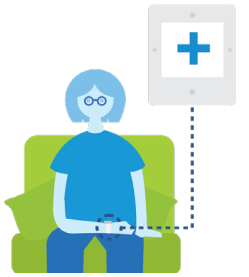
greatcall.

Before

Lacking the ability to predict incidents related to the well-being of elderly family members.

After

System leveraging past incident data combined with sensor captured information in order to **predict patients' health issues**.



Results

Ability to characterize patients according to their health status and anticipate medical emergencies.

Approach

Use data gathered by **sensors in wearable devices, as well as past incident data**, to build a system capable of predicting medical emergencies faced by elderly patients. Additionally, study patients in order to categorize them according to their overall health risk.

Technical details

The patient categorization is accomplished by grouping the users who share similar physical activity patterns and metadata such as age and gender. Several clustering and classification models are applied in conjunction with each other to define the final groups, including: logistic regression, decision trees, AdaBoost, random forest, Gaussian mixture model, and hierarchical clustering.

Once the user is characterized, the next step is to create an incident risk score by analyzing the user's physical activity of the past two weeks and demographic information. It is approached as a regression problem, where a random forest model yielded the best performance.

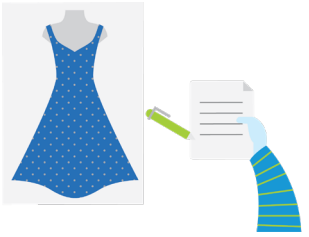
Case study 7

Copy writing product descriptions

Solution built for a large online consignment marketplace, headquartered in San Francisco, CA.

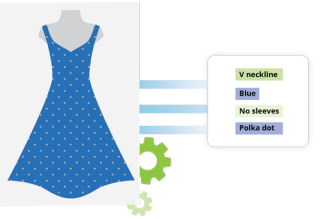
Before

Product **descriptions for unique items are written manually** in an unstructured manner. Many tedious working hours are spent, and style inconsistencies in the descriptions exist while big investments are made to train copywriters as well as edit copy for grammar.



After

Product **descriptions are generated automatically, using only product images**. The system detects key features of the items in the images and describes them accordingly.



Results

- Saving **1500 working hours** a month.
- Establishing consistent copy style.

Approach

Extract structured information from product descriptions in order to use it jointly with product images to train a deep learning model with the capability of generating **consistent product descriptions using only product item photographs**.

Technical details

Build a parser able to extract item features from unstructured product descriptions.

Create a custom deep learning model based on ResNet-50 architecture and pre-trained on ImageNet. Train the model with previously generated structured product information and photographs of the items.

The resulting model is able to classify various product attributes in one inference using only 1-3 product images per item as input data.

For production, the obtained TensorFlow model is deployed to the Google Cloud Platform (GCP) using the AI platform and cloud functions.

Case study 8

Detecting markers in pictures taken by drones

Solution built for Drone Deploy, a drone software provider, based in San Francisco, CA.

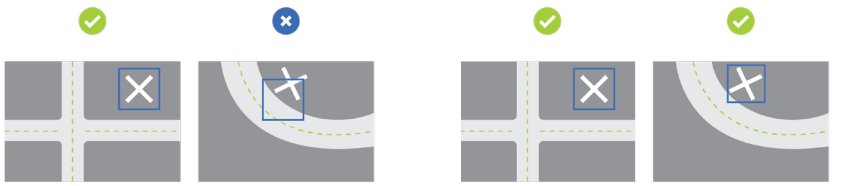
DroneDeploy

Before

Existing machine learning model created to detect markers in pictures taken by drones performs with less than 55% accuracy. Company is seeking evidence that model can be enhanced by adding deep learning techniques.

After

Proof of concept reveals that replacing the existing classification method with a system **using deep learning is a straightforward and effective solution**.



Results

- Evidence that deep learning leads to **more precise marker detection** in pictures.
- Approximating 97% accuracy and recall.

Approach

Train a computer vision model using the company's existing data sets. Prototype an **extremely accurate deep learning model for marker detection**.

Technical details

Perform object detection tasks with the help of Luminoth (a deep learning computer vision toolkit built by Tryolabs).

Leverage Luminoth's Faster R-CNN architecture, and the Regional Proposal Network (RPN) in particular, to define regions (bounding boxes) that contain markers.

Use the company's own dataset to fine-tune the checkpoints that have been pre-trained with the COCO dataset.

Case study 9

Understanding in-store customer behavior

Solution built for brick-and-mortar retailers aiming to turn video content into actionable insights.

Results

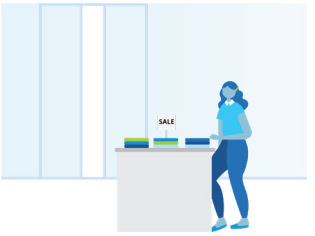
- Accumulating valuable customer data.
- Understanding customers' behavioral patterns.
- **Optimizing store layouts, storefronts, and in-store promotions based on real data**.

Approach

Create a computer vision **algorithm able to detect, track, and count people in videos** recorded by regular in-store cameras.

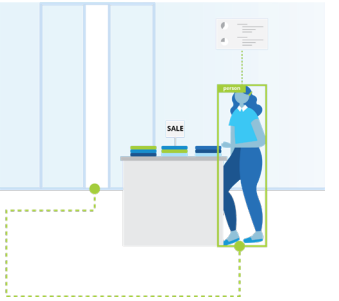
Before

Brick-and-mortar retailers have ideas about how customers behave inside a store, but lack metrics to back-up their assumptions. **Store layouts and in-store marketing campaigns aren't data-driven** and, therefore, aren't optimized.



After

Using **cameras only**, a solution is developed that reveals **insights** into customer behavior, enabling retailers to understand in-store navigation routes, hotspots, and the storefront's effectiveness.



Technical details

Experiment with various combinations of Person ReID, object tracking, pose detection, and object detection.

Notable algorithms used:

- Custom PersonLab pose detector implementation.
- Custom SORT object tracker with added EANet Person ReId for the appearance model.
- Luminoth's (a deep learning computer vision toolkit built by Tryolabs) Faster R-CNN object detector, modified for generating appearance models for objects from embeddings.

Creation of person-attention model for window fronts of retail stores. Provide accurate information, such as the number of people walking by the store, the attention they pay to the storefront, how many people enter the store, the correlation between time spent looking at the storefront and entering the store, and so on.

Case study 10

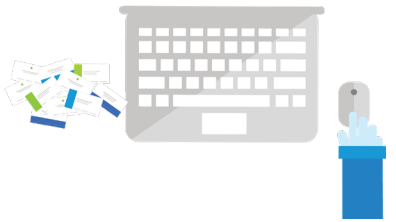
Digitizing information on business cards

Solution built for Meltwater, a SaaS company for media monitoring, headquartered in San Francisco, CA.

Meltwater

Before

Business card information is digitized and stored manually. Tedious and inefficient task.



After

App feature with the capacity to **automatically digitize and store information** present in business cards, such as name, job title, and telephone number.

Results

Efficient digitization of business card information in existing mobile app.

Approach

Development of a **feature able to detect text in business cards**, determine to what category they belong and store the information in the correct order.

Technical details

Edit images of business cards with OpenCV to eliminate noise, convert them to black and white, and improve contrast.

Pass images through the Google Vision API to perform optical character recognition (OCR) in order to obtain bounding boxes for each word and recognize its characters. Use regular expressions (regexes) to identify simple character structures like telephone numbers, social media usernames, and addresses.

Utilize a pre-trained natural language processing model from Spacy to identify complex word structures, such as names and job titles. In case a word cannot be identified, a rule-based algorithm is applied, taking into account font sizes, distances between words, and locations of the words within the frame of the business card.

About Tryolabs

At Tryolabs, we understand the challenges of completing successful machine learning projects and work closely with companies to help them in the full or partial execution of the machine learning process cycle.

We have **over 10 years of experience in building custom machine learning systems** that increase speed and quality of business processes and reveal valuable insights for strategic decision-making.

Our machine learning teams use their expertise with **techniques in the fields of computer vision, natural language processing and predictive analytics** to build and deploy systems aligned with the companies' existing infrastructures.

Given each unique business case, we provide expert machine learning services as needed, covering the following:

- Strategic implementation of custom machine learning solutions.
- Collaboration with existing machine learning teams.
- Research and prototyping.
- Machine learning training and workshops.

Would you like to know more about our work or how you can leverage data to drive outstanding results?

Connect with us via hello@tryolabs.com, we'd be happy to discuss how we can use our capabilities to transform your organization.

tryolabs.com