INSEAD
The Business School for the World®

Is it 1, 2, 3, or a chicken? Recognizing your squibbly handwriting

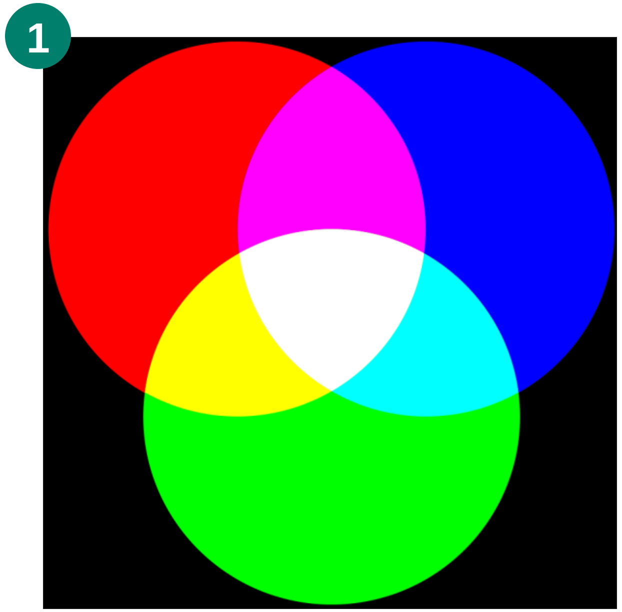
Data Science for Business Final Project: IRC4 (EA-G3)

30/05/2020

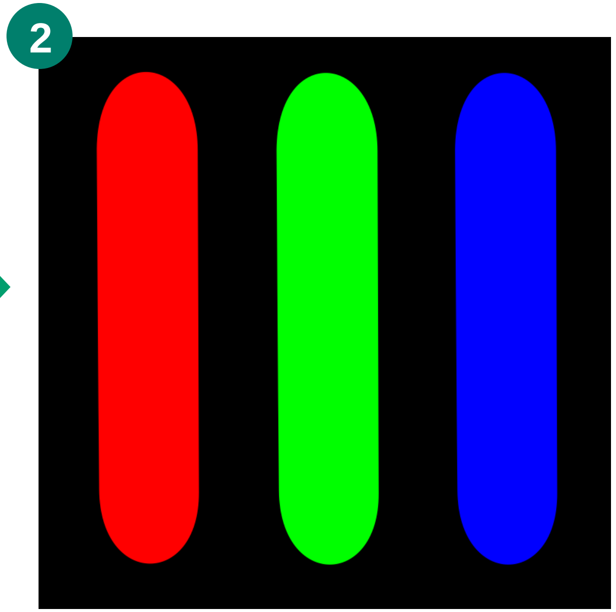
DATTAWADKAR Anay ; GUO Tingting ; MELMAN Ilia ; WEI Anais ; WIRAWAN Christian ; ZHAN Chris

Computer screens represent images by splitting each pixel into a combination of 3 different colors

INSEAD



Red, Green and Blue can be mixed to create almost any color in the spectrum



An LCD screen emulates this by splitting individual pixels into 3 “subpixels”

A 1600x900 computer screen looks like this when you zoom in -- 1600 pixels horizontally and 900 pixels vertically

Conversely, computers can "read" images by converting their component pixels into arrays of red, green, and blue values

INSEAD

R:0 G:0 B:0	R:19 G:167 B:15	R:5 G:190 B:9
R:0 G:0 B:0	R:0 G:0 B:0	R:0 G:0 B:0
R:113 G:150 B:8	R:180 G:105 B:1	R:0 G:0 B:0

RED

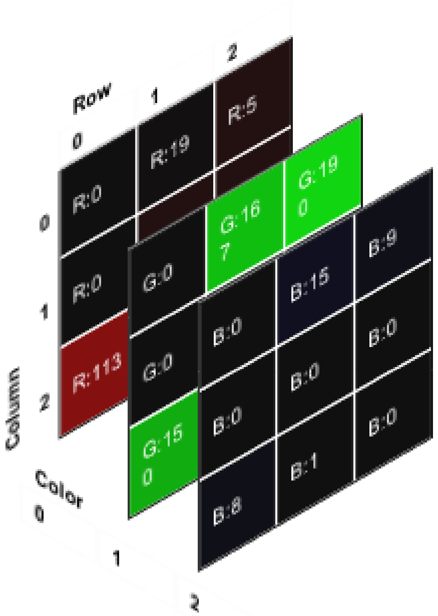
R:0	R:19	R:5
R:0	R:0	R:0
R:113	R:180	R:0

GREEN

G:0	G:167	G:190
G:0	G:0	G:0
G:150	G:105	G:0

BLUE

B:0	B:15	B:9
B:0	B:0	B:0
B:8	B:1	B:0



Computers split each pixel of an image into its respective RGB values

These pixels are then further split into 3-D arrays of code; models can then apply machine learning techniques to "recognize" and "classify" these images

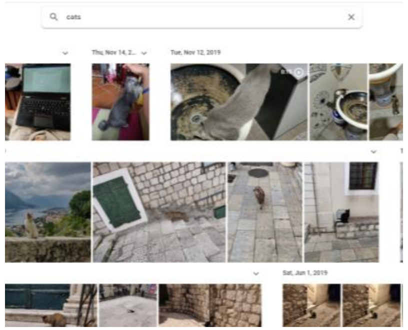
Image recognition is a rapidly-evolving AI technology with numerous current & future use cases

/NOT EXHAUSTIVE

Current Use Cases

Facial recognition

Image search



Object recognition & image detection

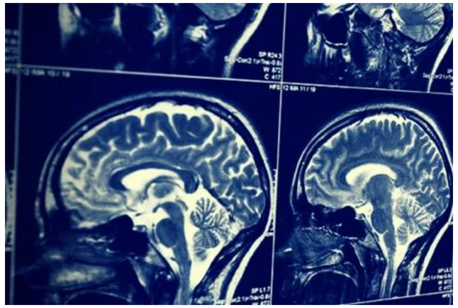
Biometrics

Augmented Reality

Potential Future Uses

Level 3 Autonomous Driving

Manufacturing Defect Identification

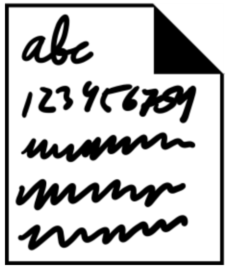


Medical imaging & disease detection

Source: Imagg

Business Problem: Optical Character Recognition (OCR) can significantly streamline a company's Accounts Payable workflow

Current Process



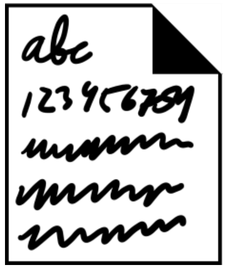
Invoices received

AP staff codes invoice amount, due date, discounts, PO number into ERP system

Invoices paid as per due date in ERP system

- Highly manual, labor-intensive, with no prioritization of invoices
- Significant lead time between invoice receipt & ERP acceptance
- Potential for manual error & unintended divergence across offices / staff members

Improved Process



Invoices received

OCR algorithm extracts key invoice info, logs in ERP; any "questionable" invoices flagged for staff review

Invoices paid as per due date in ERP system

- Significant time & labor savings
- Enables AP staff to focus on higher-value activities
- Increases traceability of process
- Difficulty when faced with new / unknown invoice data formats

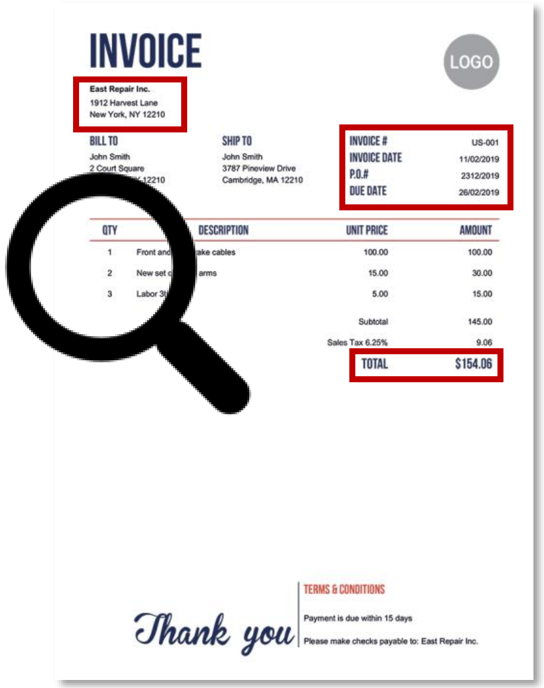
Optical Character Recognition (OCR) is a form of image recognition that enables written text to be read by a computer

1 Scan document

2 Recognize that document is invoice

3 Read the content

4 Extract key information

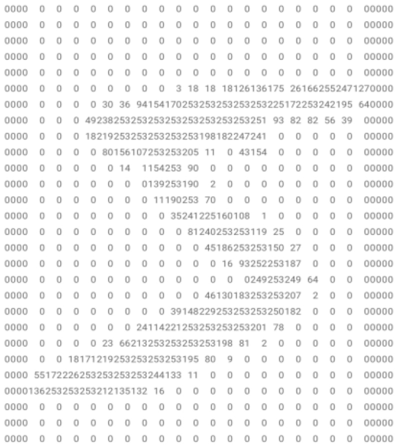


Our project is a basic OCR model that recognizes handwritten numbers

PO #:	*2312/2019*
Vendor:	*East Repair, Inc.*
Amount:	*$154.06*
Due Date:	*26/02/19*

Our project: Train machines to read numbers by applying data science classification models

Handwritten "5" as a black & white image



28x28 matrix of pixel numbers

MINIST dataset

Col [0]	Col [1]	…	Col [783]	Col [784]
5	0	…	0	0
0	…	……	……	……
1	…	……	……	……

Label to predict (Dependent variable)

Independent variables

Each matrix "flattened" to 1 row of data with 784 (=28 x28) columns; actual label appended to first 60,000 numbers (training)

Regressions & Parameters Used			
Multinomial logistic regression: expand from 1 possible outcome to 10 possible outcomes	**Random Forest**: Ntree = 100, nodesize = 50	**XGBoost**: objective=multi:softmax ; eval_metric = merror ; Cross-validation : nrounds = 10, nfold = 5	**Neural Network / Tensorflow:** Dropout rate: 0.4 Batch size: 512

Results: Confusion matrix comparison and mismatches

Actual Class \ Predicted Class	0	1	2	3	4	5	6	7	8	9
0	934	0	6	4	0	15	12	6	3	0
1	0	1109	6	4	1	1	4	1	8	1
2	6	13	921	19	9	10	9	13	28	4
3	4	4	23	898	2	32	2	15	20	10
4	3	3	9	5	896	2	14	8	10	32
5	12	3	5	37	6	758	19	5	41	6
6	12	7	18	2	6	23	883	3	2	2
7	2	12	17	9	9	1	1	937	5	35
8	9	9	11	22	8	29	14	14	847	11
9	6	8	1	10	29	6	3	29	14	903

Multinominal logistic regression
90.86 %

Prediction \ Reference	0	1	2	3	4	5	6	7	8	9
0	1161	0	5	4	4	5	6	3	3	8
1	0	1312	1	0	1	2	2	8	4	2
2	0	9	1143	20	6	2	1	19	11	5
3	0	4	10	1140	0	11	0	0	16	17
4	1	4	6	2	1113	3	3	7	7	17
5	1	3	0	18	0	1030	14	0	8	7
6	6	0	7	0	2	10	1147	0	8	0
7	0	3	7	12	4	1	0	1187	0	13
8	16	8	11	22	9	11	11	6	1101	17
9	0	5	2	8	29	9	0	23	12	1104

Random forest
95.32%

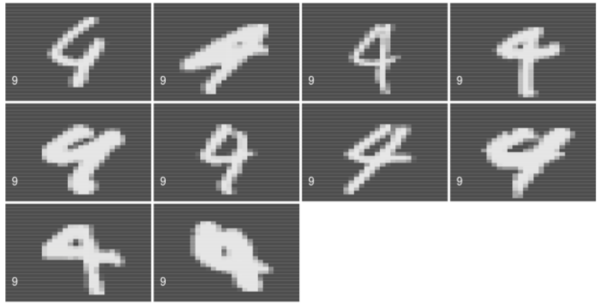
Prediction \ Reference	1	2	3	4	5	6	7	8	9	10
1	1325	4	8	4	3	3	6	28	3	0
2	5	1122	32	7	4	1	5	11	6	0
3	2	17	1111	0	29	0	5	8	10	0
4	4	10	4	1106	6	5	9	7	31	0
5	1	0	15	1	991	23	1	16	3	0
6	0	4	5	5	13	1130	0	9	1	0
7	5	14	7	2	4	0	1197	2	15	0
8	4	8	30	5	15	14	2	1068	6	0
9	2	4	10	37	16	0	26	17	1109	0
10	0	0	0	0	0	0	0	0	0	0

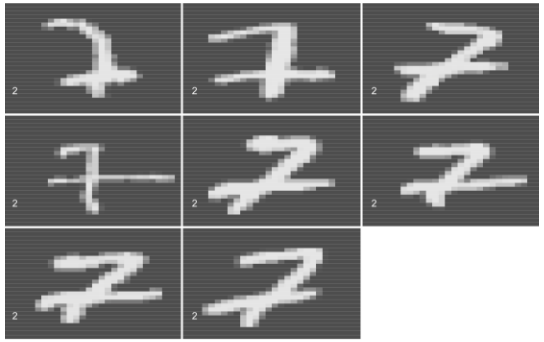
XGBoost
94.26%

Predicted \ Actual	0	1	2	3	4	5	6	7	8	9
0	971	0	5	0	1	2	6	1	4	2
1	1	1126	1	0	0	0	2	4	0	2
2	1	3	1012	2	5	0	0	7	4	0
3	0	0	1	996	0	9	1	1	6	4
4	0	0	1	0	960	1	4	1	1	4
5	1	0	0	3	0	871	4	0	2	2
6	3	2	2	0	3	4	941	0	1	1
7	1	1	6	3	1	0	0	1007	2	4
8	2	3	4	3	1	3	0	1	951	0
9	0	0	0	3	11	2	0	6	3	990

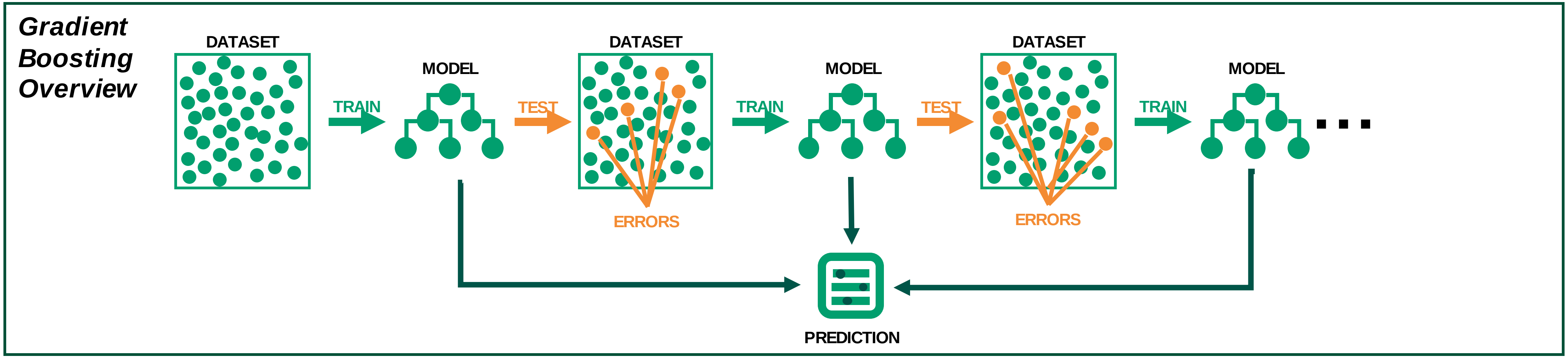
Neural network
98.26%

What are the 1.74% Prediction Errors?
4s & 9s
2s & 7s





XGBoost models make predictions using an advanced form of gradient boosting



Ensemble:
Collection of predictors used to make a final prediction

Bagging: Independent predictors, averaged (e.g., Random Forest)

Boosting: Predictors made sequentially (e.g., Gradient Boosting)

Code Example

```r
# convert every variable to numeric, even the integer variables
train <- as.data.frame(lapply(train, as.numeric))
cv <- as.data.frame(lapply(cv, as.numeric))

# convert data to xgboost format
data.train <- xgb.DMatrix(data = data.matrix(train[, 2:ncol(train)]), label = train$label)
data.cv <- xgb.DMatrix(data = data.matrix(cv[, 2:ncol(cv)]), label = cv$label)

watchlist <- list(train  = data.train, test = data.cv)

parameters <- list(
    # General Parameters
    booster            = "gbtree",          # default = "gbtree"
    silent             = 0,                 # default = 0
    # Booster Parameters
    eta                = 0.3,               # default = 0.3, range: [0,1]
    gamma              = 0,                 # default = 0,   range: [0,∞]
    max_depth          = 6,                 # default = 6,   range: [1,∞]
    min_child_weight   = 1,                 # default = 1,   range: [0,∞]
    subsample          = 1,                 # default = 1,   range: (0,1]
    colsample_bytree   = 1,                 # default = 1,   range: (0,1]
    colsample_bylevel  = 1,                 # default = 1,   range: (0,1]
    lambda             = 1,                 # default = 1
    alpha              = 0,                 # default = 0
    # Task Parameters
    objective          = "multi:softmax",   # default = "reg:linear"
    eval_metric        = "merror",
    num_class          = 10,
    seed               = 1234       # reproducability seed
    )

xgb.model <- xgb.train(parameters, data.train, nrounds = 10, watchlist)
xgb.predict <- predict(xgb.model, data.cv)
print(xgb.cm <- confusionMatrix(factor(xgb.predict,levels = 1:10), factor(cv$label,levels=1:10)))
```

Neural Networks use simulations of the human brain to make predictions

What is an Artificial Neural Network (ANN)

- An ANN is a simulation of many densely interconnected “cells” inside a computer, modelled after neurons in the human brain
- This network can learn things, recognize patterns, and make decisions in a humanlike way.
- Neural networks can solve prediction or classification problems

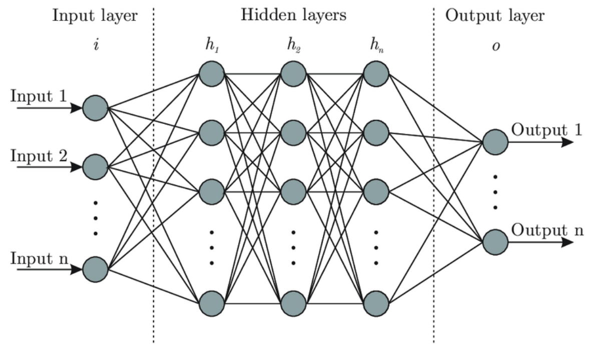
When should you use one?

- Neural networks are overkill for many applications; simple regression will usually do the trick
- Neural networks are flexible models, and can “pick” the best type of regression for a given dataset
- They are, as a result, more complex and computationally intensive.

What is TensorFlow

- An open-source software library for numerical computation using data flow graphs
- Commonly used for deep learning, it also includes algorithms for as K-Means clustering, random forests, Support Vector Machines, Linear/Logistic Regression, and Gaussian Mixture Model clustering

Examples




```
# common optimizers: https://keras.io/optimizers/
# adam -- commonly used
# SGD -- "stochastic gradient descent"

# common metrics: https://keras.io/metrics/
# accuracy, mae, mape

model %>% compile(
  loss = 'categorical_crossentropy',
  optimizer = 'adam',
  metrics = c('accuracy')
)

# Training / "fitting" the model

history <- model %>% fit(
  x_train, y_train, # on what data to train
  epochs = repetition, # how many repetitions to have
  batch_size = bs, # how many datapoints are fed to the network at a time
  validation_split = 0.2  # percentage of training data to keep for cross-validation
)

#summary(model)

#plot(history)

model %>% evaluate(x_test, y_test) # apply the model to testing data
```

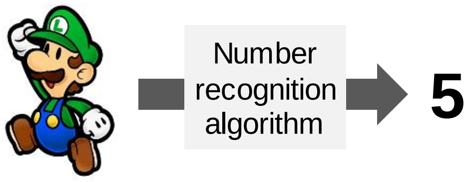
Go-to choice of real-world image recognition applications

Despite great advances, today's models still face significant limitations, and there is a long way to go

INSEAD

Computers lack "common sense"

- Computers don't know when to say "I don't know"
- As such, even when presented with spurious data, the algorithm will attempt to make a prediction based on its training
- This "overconfidence" can have inaccurate / comical results

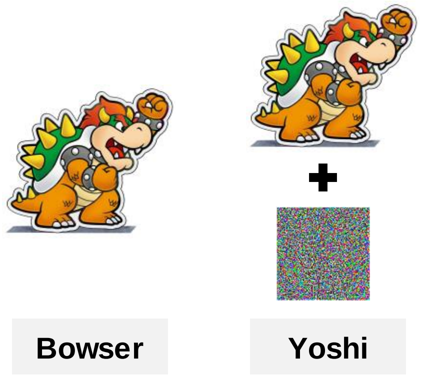


Difficulties with model generalization

- Models are trained and evaluated by randomly splitting data into "training" and "testing"
- Real-world data can differ in viewing angles, scene configurations, camera quality, etc.
- When applied to circumstances beyond their training, models become increasingly inaccurate

Potential for malicious attacks

- People have been able to "fool" image recognition models by using carefully-constructed "noise"
- So-called "adversarial attacks" are model-agnostic, enabling easy "transferability" across a range of algorithms and use cases
- This raises significant security concerns



Comprehensive scene understanding

- Algorithms today can identify objects & groups within a scene (perception)
- However, they struggle to understand object relationships (what is actually happening)
- As such, they cannot yet form a cognitive understanding of the physical world

INSEAD

The Business School
for the World®

Europe | Asia | Middle East