

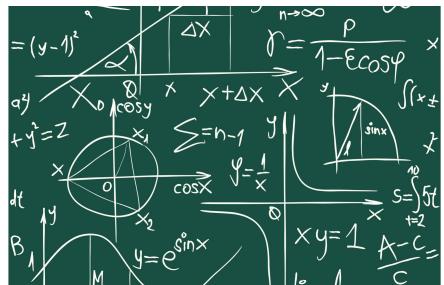
eCommerce & Artificial Intelligence (AI)

Bill Gold

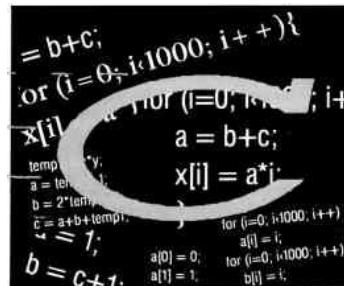
Principal Ataeva Consulting

April 2017

Introduction - Bill Gold - Principle Ataeva



+



+



Data Science

- 200+ Algorithms
- Commercial
- Patented

Technology

- Commercial S/W
- 2MM Lines of Code

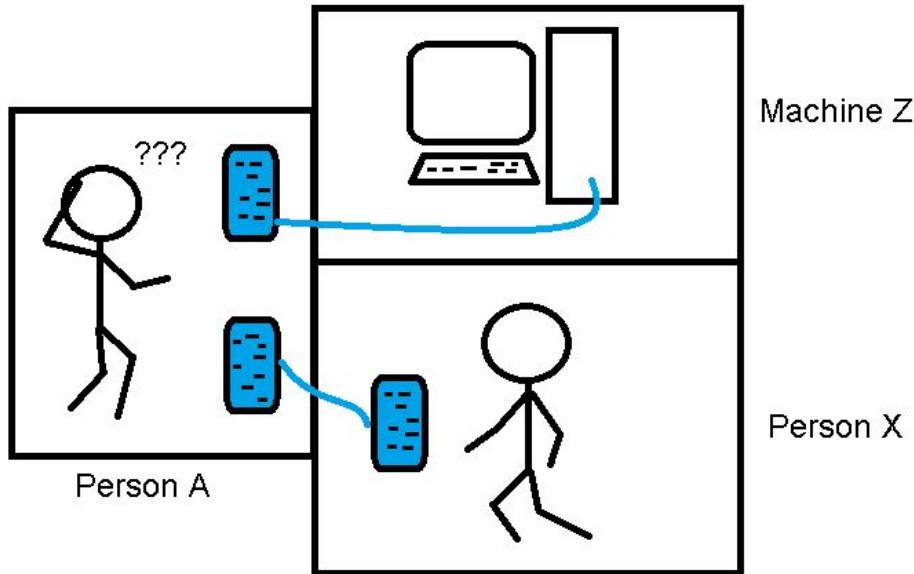
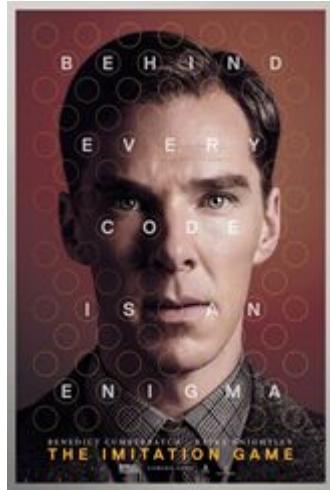
Strategy Consultant

- \$500 MM+ ROI
- Product Development
- Trusted Advisor

Agenda

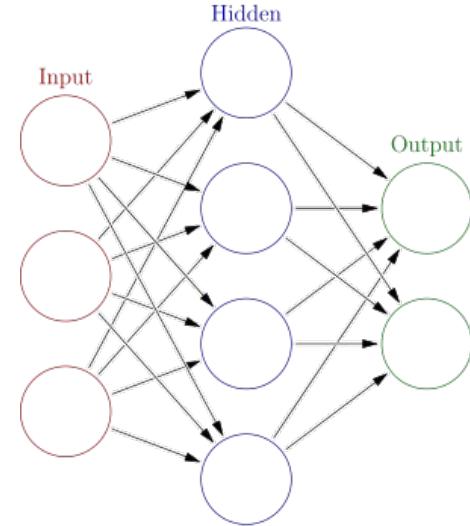
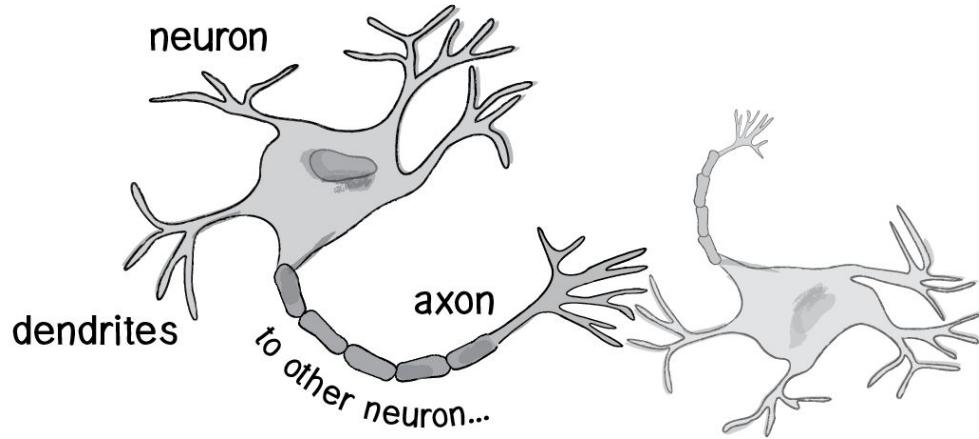
		~Minutes
I.	What is AI?	10
II.	A Business View of AI	15
III.	eCommerce and AI	15
IV.	Q & A	5
	Sample Model (Time Permitting)	45 Total

I. What is AI? The Turing Test



Neural Networks, A Brief Look Back

Computers inspired by brain neurons



Lots of Research. Little Impact.

Key AI Milestones

Chess



2 Moves → 400 Options
1997
Garry Kasparov vs. IBM
Brute Force Method

Go



2 Moves → 13,000 Options
2016
Lee Sedol vs. Google
Deep Learning, Reinforcement

Why Now? Confluence of Events

Globalization	Fiber Optic Cables, Shift to Capitalism, Economic Bubbles
Analytics	25 years of computer applications, commodity hardware
eCommerce	PC, Internet, Mobile Devices
Artificial Intelligence	New Techniques, GPU, Unstructured Data

Data Augmentation

Original Picture



1 raw

- Normalize
- Crop
- Zoom
- Rotate
- Blur
- Brighten



19 augmented

II. AI, A Business View

Deep Learning

Why is this method different from all other methods? Deep Learning excels at:

- Detecting Hierarchies
- Finding Signals in Unstructured Data

Unstructured Data

Visual	Edge, Wheel, Car Person, Gender, Individual	Self Driving Cars MRI Diagnostics
Language	Related Words Specific to Generic	Chatbots Customer Service
Voice	Syllable, Word, Phrase, ... Smaller to Larger	Smart Assistants Customer Service



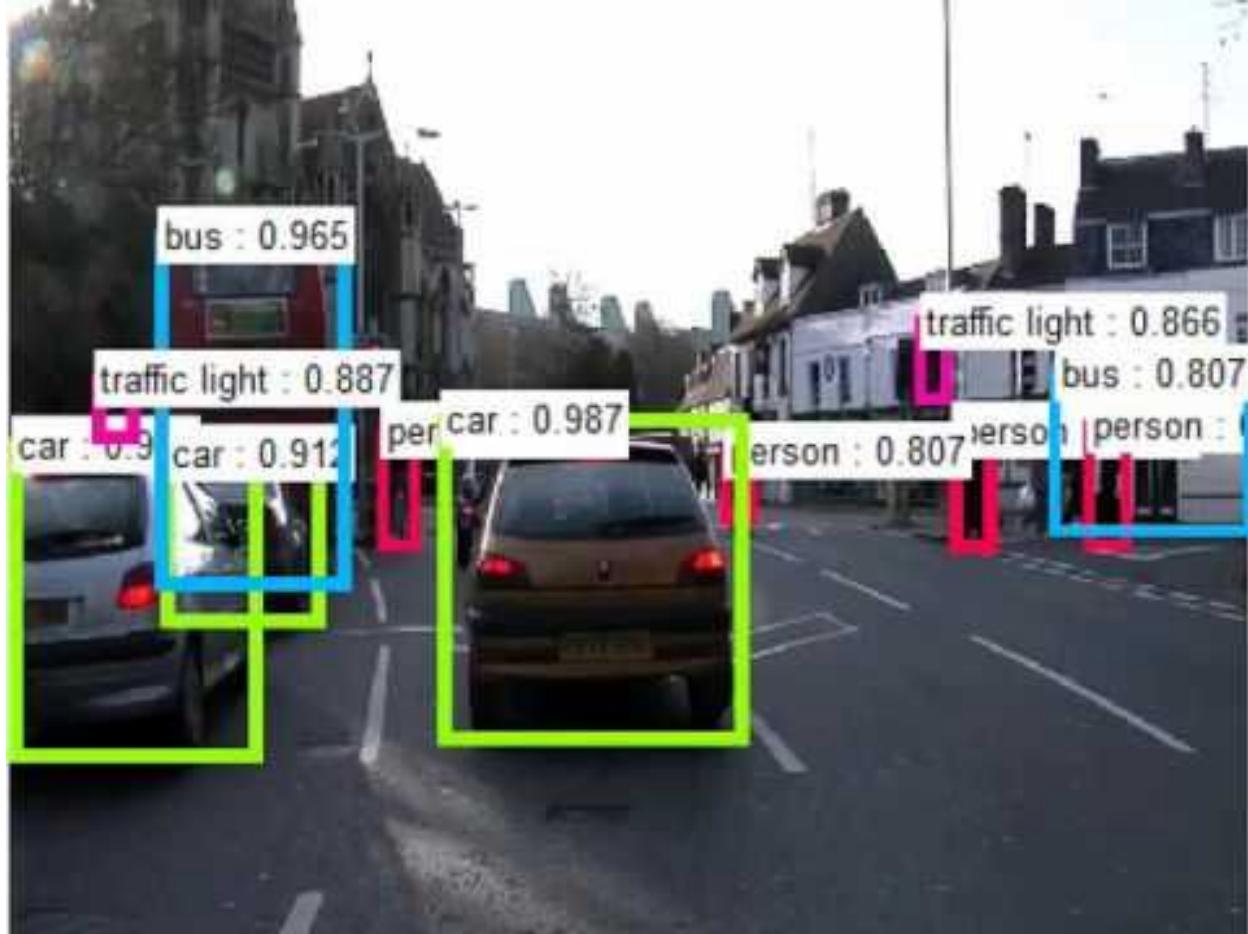
Global Smartphones

2015	2.6	B
2020	6.1	B

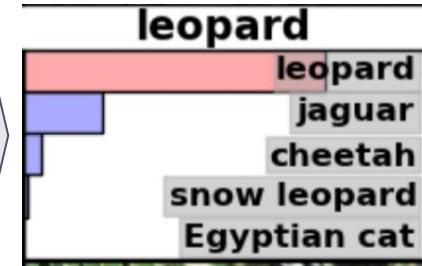
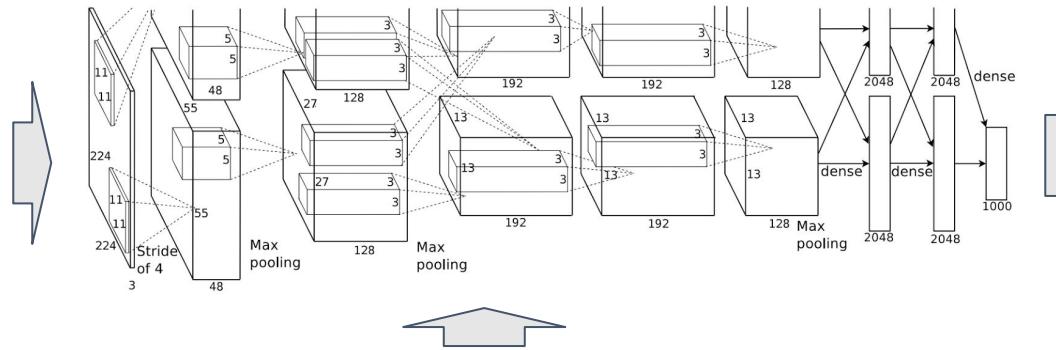
Visual

Recurrent
Convolutional
Neural Network

RCNN

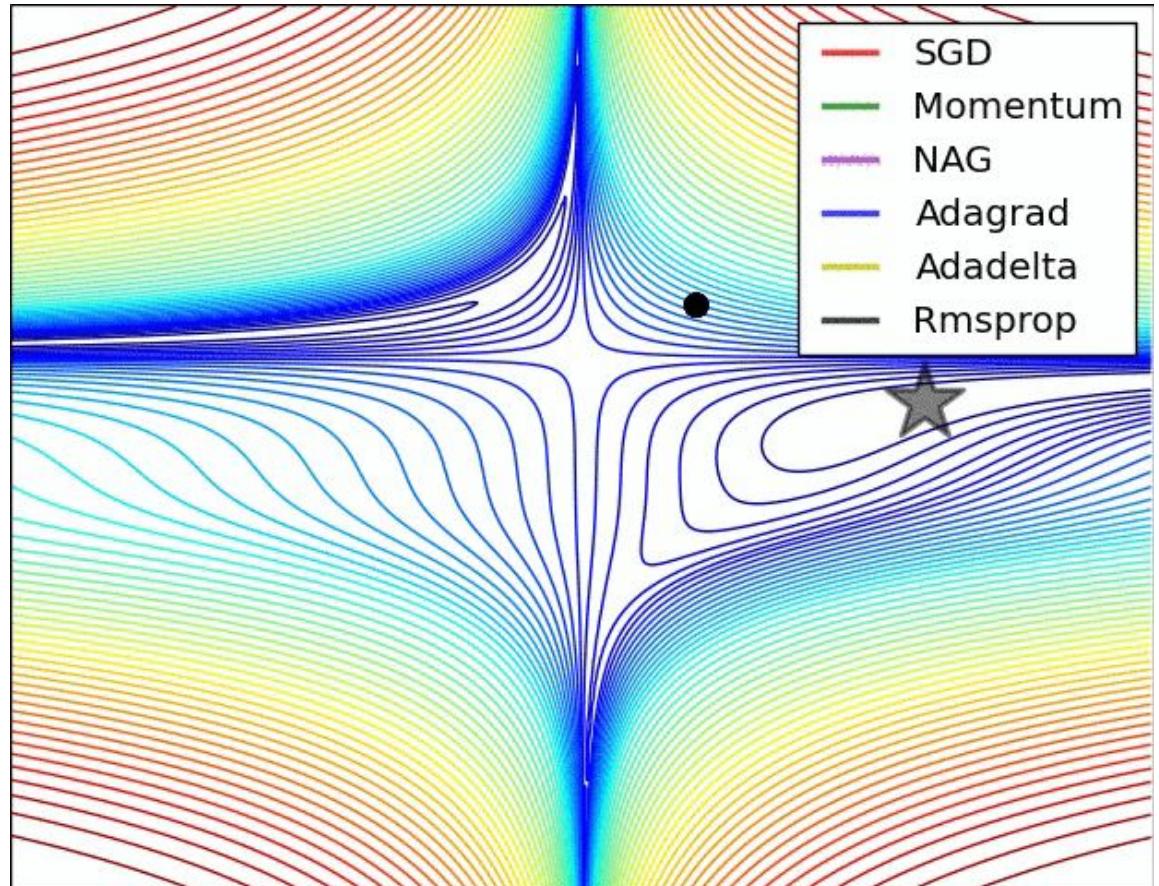


Deep Learning



Learning

Back Propagation



What is the Optimal Location?

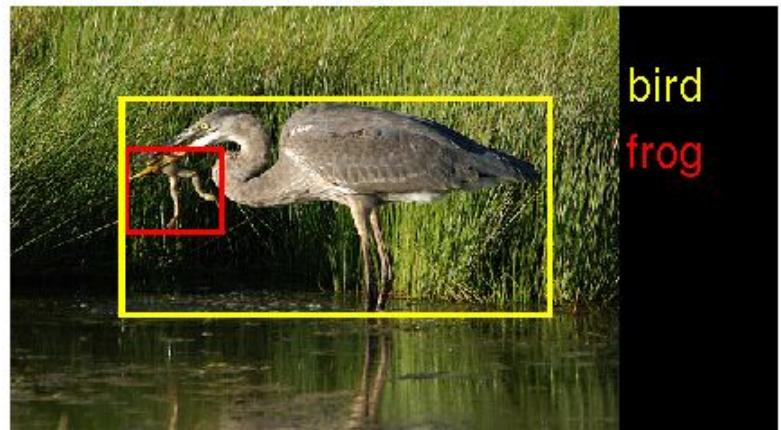
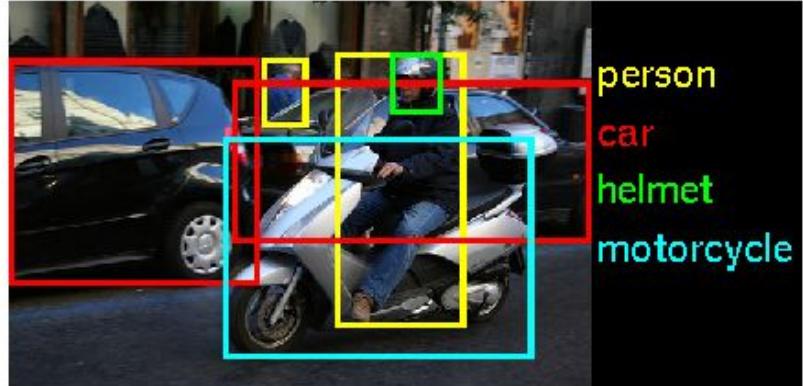
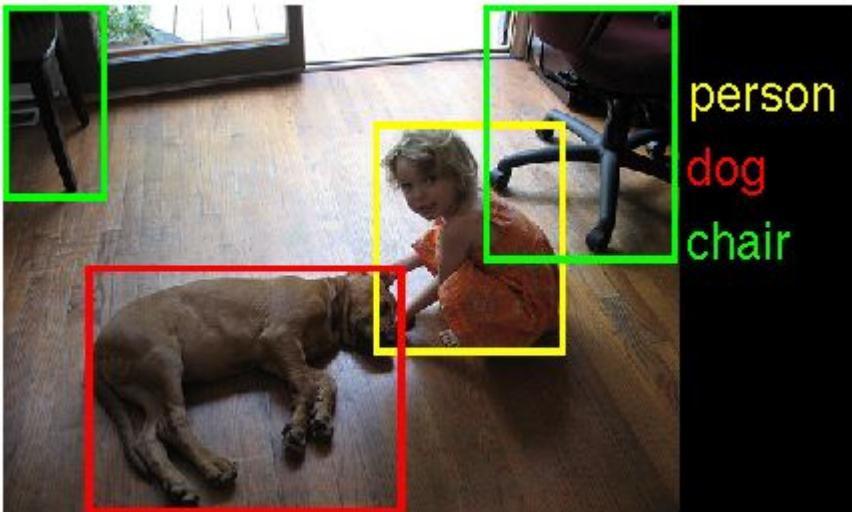
A mathematical measure of truth labels

- Is the MRI Mass Cancerous?
- Should the car turn?
- What is the spoken word?

III. AI & e Commerce

1. Case Studies
2. Sample AI Model

Truth Labels



Thousands to Millions

Case Study



Input: Pixels

Output:

Wine Identified
Prices
Offers

86% Match Rate

Uses CNN + OCR¹

12 MM+ Users

Case Study

ViSENZE

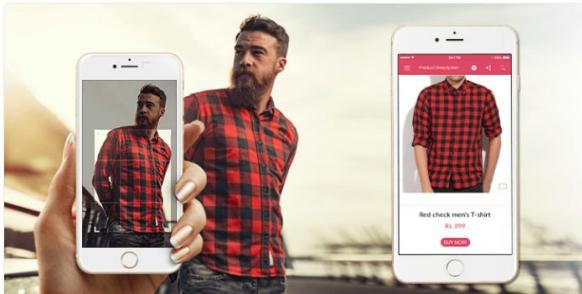


Input: Pixels
Output:
Voice & Image Search
for eCommerce
Raised \$54M
in 2016

Case Study



Search



Visual Search

We utilize state-of-the-art deep learning technologies to provide precise reverse image search solutions. The search engine is designed in a way that it can retrieve exact matches from the indicated database for an input image. Various information like content, color, shape and texture are extracted and fused together to provide visual search solutions for various e-commerce businesses in the field of fashion, housing, medicine etc.

Upsell



Complete Look Recommendation

Confused about what earrings should be paired with that "Sari"? Our AI powered engine processes thousands of looks to understand and learn how to pair accessories and apparel with each other to provide customers the best holistic shopping experience they could want.

Recommender



Visual Recommendation

Tired of using the traditional "Users also viewed this!" Try the intuitive and visual data based recommendation engine developed by top researchers and engineers at Staqu. With an aim of providing relevant recommendations, our algorithms sorts products based on popularity and visual content to provide a much more human touch to your shopping experience.

Case Study



Input: Pixels
Output:
Voice & Image Search
for eCommerce
Raised \$18M
in Q1'16

Case Study



Walk In Scan



Just Walk Out

How might Amazon Go's technology work?

Who has taken an item?



Bluetooth beacons can identify whose mobile device is nearest the shelf. A dense beacon network can be accurate to within less than 0.5 metres.

What item was taken?

Shelf cameras will detect when an item has been removed or added and what that item looked like. This data feeds into an AI system.



Artificial Intelligence will look at vision, weight and stock location data to make best guess of what item(s) have been added or removed



Shelf weight sensors will likely be used to detect the weight of items removed or added. This data feeds into an AI system.

The Amazon Go app in the shopper's mobile device will be communicating with the store's beacon network.



Just Walk Out

Trends to Watch

- Chatbots
- eCommerce inside retail stores
- Mining in store video
- Next generation Webcam

Ideas That Excite Me

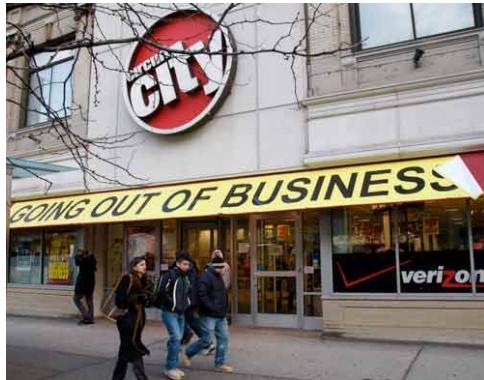
- Summarize a 30s Video Clip
- Buying Glasses & Clothing

IV. Q&A

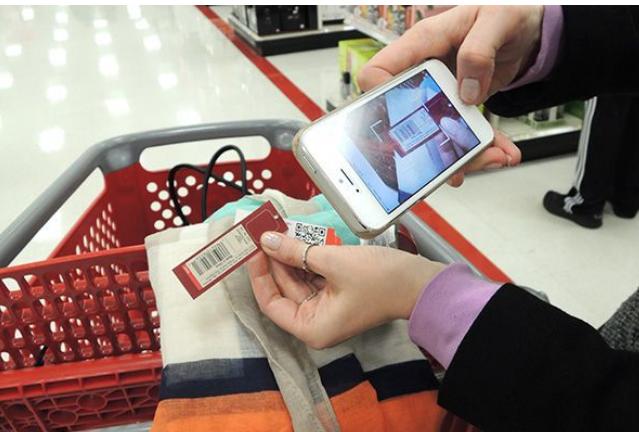


Sample AI Model - In Store Shopping

1. Retail Stores Struggle...



2. Current Solutions



Retail Innovation 1936



The Shopping Cart ¹

- Invented On: June 4th 1936
- Inventor: Sylvan Goldman
- Design Inspiration: ???
- Benefit ²: ???

Source: 1 - Washington Times <http://www.washingtontimes.com/news/2011/sep/12/great-store-put-in-shopping-cart-quality/>

2 - Paco Underhill - "Why We Buy: The Science of Shopping"

Key Benefits

To the Consumer

- Faster Checkout
- Relevant Offers
- Time Savings
- Social Media Shopping
- Product Safety Info

To the Retailer

- Sales Increase
- Real Time Marketing

Success Criteria

<ul style="list-style-type: none">• Convenience	What do the 4 current solutions have in common?
<ul style="list-style-type: none">• High accuracy	40k unique products ¹
<ul style="list-style-type: none">• Low latency	50-70k @Wegmans ²
<ul style="list-style-type: none">• Time to Market	FAST!!!

1 - Food Marketing Institute - <http://www.fmi.org/research-resources/supermarket-facts>

2 - Wegmans - https://www.wegmans.com/webapp/wcs/stores/servlet/CategoryDisplay?storeId=10052&identifier=CATEGORY_2441

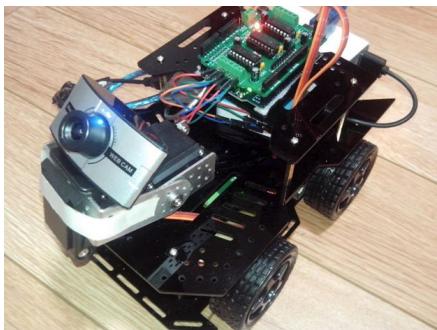
3. Solution Architecture

Phase	Prototype Status	Beta Status
A. Product Architecture	✓	Pending
B. Data + Image Collection	✓	On Going
C. Image Augmentation	✓	Pending
D. Training & Validation	✓	Pending
E. Deployment	WIP	Pending
F. Next Steps	n/a	n/a

A. Product Architecture

Inventory

*



Product Detection

*



Internal GPS

Recommendation Engine

Note * - Modules I Designed & Built Vision Components

B. Capturing Baby Product Images

Colleagues & Friends

- In Store Pictures
- By colleagues +
- Manual Truth Labels
- ~500 Today
- On Going

Internet

- I Designed Data Tables SQLite
- Freelancer Hired to Crawl Python
- 20k Unique Pictures Captured
- Automated Truth Labels

Data Table View

	A	B	C	D	E	F	G	H	I	J
1	Table Description	BrandsRaw Contains one row for each Brand e.g. Abbott, Semilac etc..								
2	Field Name	BrandID	BrandNa	BrandNameRefined	BrandParentID	prdBatch	prdBatchDt	TypeCd	StatusCd	
	Sample Value	1	Abbot	Abbott	7	1	1/11/2016 0:00:00			
8	Field Definitions	Field Name	Unique	Field Source	Type	Valid Value	Field Description	create table BrandsRaw(
9	BrandID	Y	Calculated	integer	Not blank	A calculated (derived) hash tag which uniquely identifies the product. e.g. Similac Go & C			BrandID integer,	
10	BrandNameRaw	Y	Scrape	varchar(128)	Heinz	Company which sells the product. With only [A-z,0-9] this name will be used in sub-directories			BrandNameRaw varchar(128),	
11	BrandNameRefined	N	Calculated	varchar(128)		Used to standardize Brand name (eg. Abbott Labs and Abbot)			BrandNameRefined varchar(128),	
12	BrandParentID	N	Calculated	integer		Used for identifying Brand Hierarchies. For example ABBOTT owning Semilac is a parent			BrandParentID integer,	
13	prdBatch		TBD	bigint		An integer identifying a batch of data which has been captured			prdBatch bigint,	
14	prdBatchDt		TBD	varchar(32)		Date and time batch of data was captured			prdBatchDt varchar(32),	
15	TypeCd			varchar(8)		BRAND - Branded on Label, RET - Retail (Not on label)			TypeCd varchar(8),	
16	StatusCd			varchar(8)		ACTV - Active, HOLD			StatusCd varchar(8);	
17										
18										
19										
20										

3

4

Data Table View

1

2

4

3

A	B	C	D	E	F	G	H	I	J	K
1	Table	PrdImgRaw								
2	Description	Contains one row for relevant images on a product web page. For example the URL below will create (8) rows one for each of the larger product images towards the upper left hand side of the web page. Video:								
4	Sample URL	http://www.amazon.com/Similac-Formula-Powder-Frustration-Packaging/dp/B013BM4X5G/ref=lp_16323111_1_1s=baby-products&ie=UTF8&qid=1450811207&s=1-1								
5	Field Name	prdID	prdSource	prdURL	prdBrand	prdName	imgNameRaw	imgNameRefine	imgPath	imgDimensions imgWidth
6	Sample Value	YT2DE4R	Amazon	http://www.amazon.com	Similac	Similac Go & Grow+ Stage 1	71BiJt4BhAL__SX425_.jpg	0000000_000001	\Images\Smilac\	
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										

In Store View



Web Crawler View



- Brand - Abbot
- Product - PediaSure Chocolate
- Product Size - 237 ml
- UPC Visible - N
- QRC Visible - N

D. Image Augmentation

What image augmentations
are relevant to an in store
shopping experience?

Augmentation Methods

- Zoom
- Rotate Spoiler Alert...
- Blur Camera Filters Are Significant Too
- Brighten

Augmentation Example



```
408 #####
409 #### Transfrom all images in sDB
410 #####
411 #####
412 sBrandsToProcess = enumerateBrands ()
413
414 for k, prdBrand in enumerate(sBrandsToProcess):
415
416     # sprdBrand = prdBrand.replace(' ','')
417     con = sqlite3.connect(sDB)
418
419     curDel=con.cursor()
420     sQuery = ("DELETE      " +
421               "FROM        ImgTrans " +
422               "WHERE      prdBrand = '%s'" % prdBrand)
423
424     # curDel.execute('DELETE FROM ImgTrans WHERE prdBrand == ?', (prdBrand,))
425     curDel.execute(sQuery)
426     con.commit()
427     curDel.close()
428
429     curSel = con.cursor()
430
431     sQuery = ("SELECT      *      " +
432               "FROM        PrdImgRaw " +
433               "WHERE      prdBrand = '%s' " % prdBrand +
434               "      AND      imgTypeCd = 'SING' " +
435               "      AND      imgStatusCd = 'T' " +
436               "ORDER BY    prdBrand, imgPath ASC" )
437
438     print (sQuery)
439     curSel.execute (sQuery)
440     rows = curSel.fetchall()
441
442     for row in rows:
443         # print (sRawImagePath , row[7])
444
445         curIns = con.cursor()
446         prdImageOrig = Image.open(sRawImagePath + row[7])                                # Original Image
447         CreateOutputDir (row[6],row[7])
448         prdImageNorm = imgNormalizeFix(prdImageOrig, row[6], row[7], row)                # Normalize Image
449         prdImageCrop = imgCrop(prdImageNorm, row[6], row[7], row)                         # Crop Image
450         imgZoom (prdImageCrop, row[6], row[7], row)                                         # Zoom Image
451         imgRotate (prdImageCrop, row[6], row[7], row)                                     # Rotate Image
452         imgBlur (prdImageCrop, row[6], row[7], row)                                       # Blur Image
453         imgBrighten (prdImageCrop, row[6], row[7], row)                                 # Brighten Image
454
455     con.commit()
```

Augmentation Outputs

Original Picture



1 raw

- Normalize
- Crop
- Zoom
- Rotate
- Blur
- Brighten



19 augmented

E. Training & Validation

- i. Analysis of Images
- ii. Split Data 70/5/25
- iii. Network Design, Start with Alexnet
- iv Pretraining Tests
- v Training - Issues, Solutions and More Training
- vi Mobile Accuracy and Latency
- viii Next Steps

i. Analysis of Images

Raw Counts Of

- Brands - 985
- Products - 12,542
- Images - 20,775

```
1 install.packages("RSQLite")
2 install.packages("data.table")
3
4 library("RSQLite")
5 library("data.table")
6
7 sOutputPath <- "/home/billgold/SmartCart/ProtoType/v003/TransformedImages/"
8 protoDB <- "/home/billgold/SmartCart/ProtoType/v003/proto"
9 conn <- dbConnect(drv = SQLite(), dbname= protoDB)
10
11
12 ### How Many Brands, Products & Images?
13
14 q <- paste ('SELECT      count (DISTINCT prdBrandNameRefined), ',
15             '              count (DISTINCT prdID) ' ,
16             'FROM          PrdRaw ', sep=' ')
17 dbGetQuery(conn, q)
18
19 q <- paste ('SELECT      count (DISTINCT prdBrandNameRaw), ',
20             '              count (DISTINCT prdBrandNameRefined), ',
21             '              count (DISTINCT prdID), ',
22             '              count (DISTINCT imgNameRefined) ' ,
23             'FROM          PrdImgRaw ', sep=' ')
24 dbGetQuery(conn, q)
25
26
```

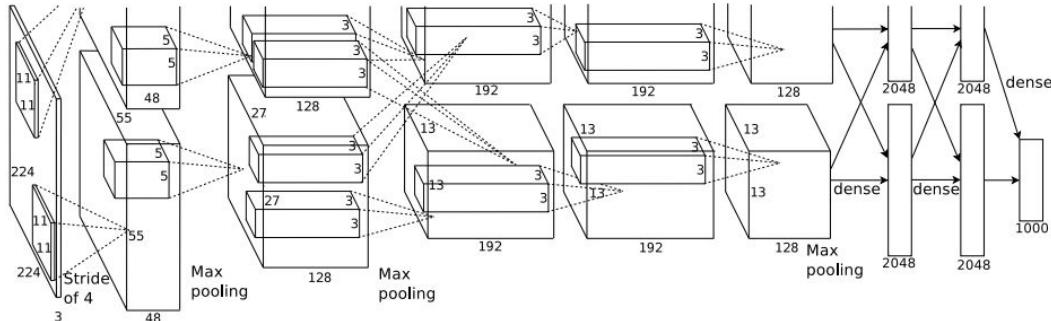
ii. Split Images

- 70% Training
- 25% Validation
- 5% Test
- Output to DIGITS

```
1 library("RSQLite")
2 library("data.table")
3
4 sOutputPath <- "/home/billgold/SmartCart/ProtoType/v003/"
5 protoDB <- "/home/billgold/SmartCart/ProtoType/v003.proto"
6 conn <- dbConnect(drv = SQLite()), dbname= protoDB)
7
8 #####
9 #### Create table BrandsSelected
10 #####
11 q <- paste("SELECT pir.prdBrandNameRefined, count(DISTINCT pir.prdID) as prdIDCnt, count (DISTINCT pir.imgNameRefined) as imgNameCnt",
12 "FROM PrdRaw pr, PdImgRaw pix, BrandsRaw br",
13 "WHERE pr.prdID = pir.prdID",
14 "AND pir.prdBrandNameRaw = br.BrandNameRaw",
15 "AND br.TypeCd != 'RET' ",
16 "AND pir.imgTypeCd = 'SING' ",
17 "GROUP BY pir.prdBrandNameRefined",
18 "HAVING count(DISTINCT pir.prdID) >= 20",
19 "sep='\\n')
BrandsSelected <- data.table (dbGetQuery(conn, q))
BrandsSelected$prdBrandNameRefined
BrandsSelected$Class <- 0:(nrow(BrandsSelected)-1)
View (BrandsSelected)
dbWriteTable(conn = conn, name = "BrandsSelected", value = BrandsSelected, row.names = FALSE)
25
26
27 #####
28 #### Training Validation Split of Images
29 #####
30 seed <- as.integer ( as.double ( Sys.time() ) * 1000 + Sys.getpid() ) %% 2^31
31 cat (seed) # 872720482
32 set.seed ( 872720482 )
33
34 nrows <- 20775
35 Pct70 <- as.integer ( nrows * 0.7 ) + 1
36 Pct25 <- as.integer ( nrows * 0.25 ) + 1
37 Pct05 <- as.integer ( nrows * 0.05)
38 Pct70
39 Pct25
40 Pct05
41 Pct70 + Pct25 + Pct05
42
113 #####
114 #### Create Class file for DIGITS
115 #####
116 #### Create list of Brands for python code to transform images for..
117 cat ( paste ('',BrandsSelected$prdBrandNameRefined,"",sep=""),sep="\n")
118
119 #### Create DIGITS Brand (class) file
120 dt1 <- BrandsSelected [ ,c('prdBrandNameRefined','Class'),with=F]
121 write.table(dt1, paste(sOutputPath,"BrandsSelected.txt",sep=""), row.names=FALSE, col.names = FALSE, quote=FALSE, sep="\t")
122
123
124
125
126 dbDisconnect(conn)
127
```

iii. Network Design Version 01

Alexnet¹



Hyperparameters

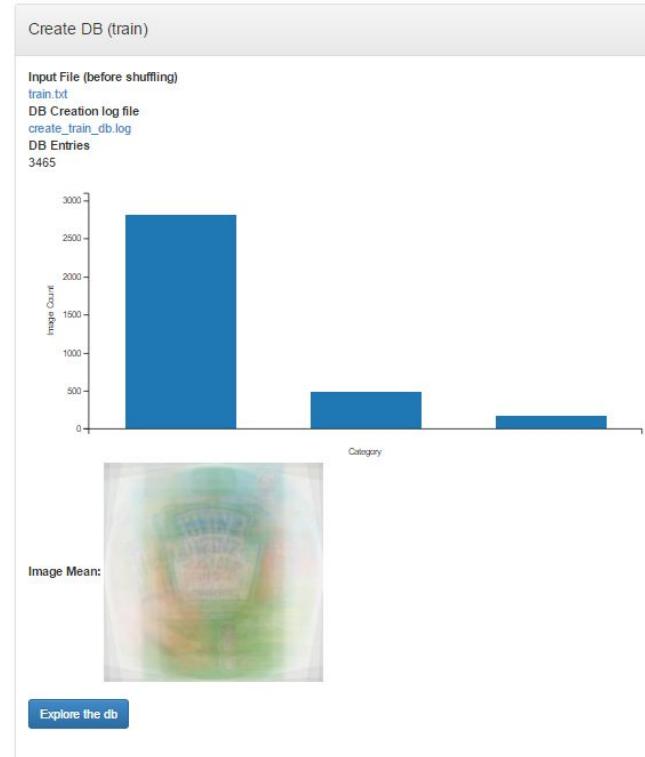
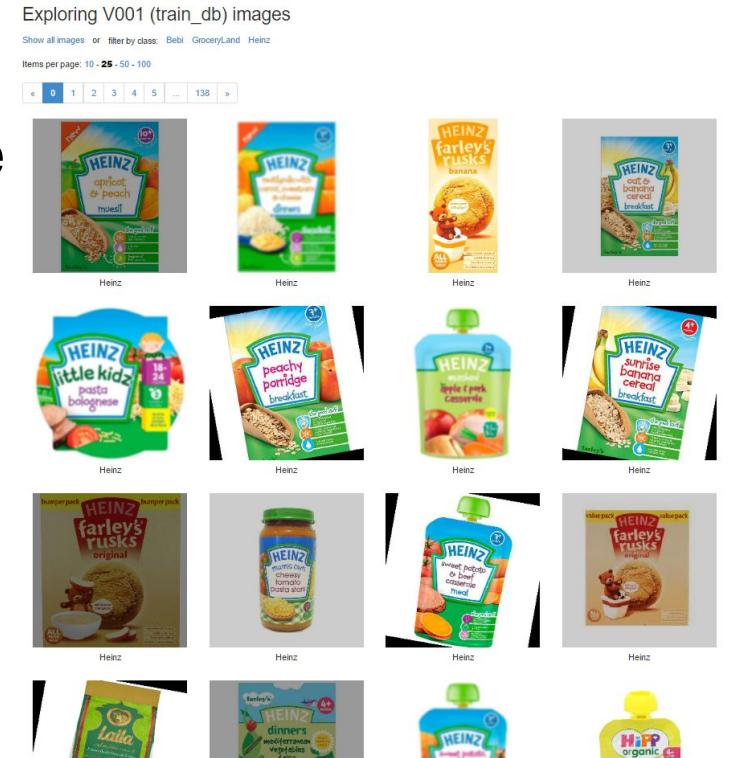
```
test_iter: 12
test_interval: 35
base_lr: 0.01
display: 4
max_iter: 1050
lr_policy: "step"
gamma: 0.1
momentum: 0.9
weight_decay: 0.0005
stepsize: 347
snapshot: 35
snapshot_prefix: "snapshot"
solver_mode: GPU
net: "train_val.prototxt"
solver_type: SGD
```

iv. Small Tests

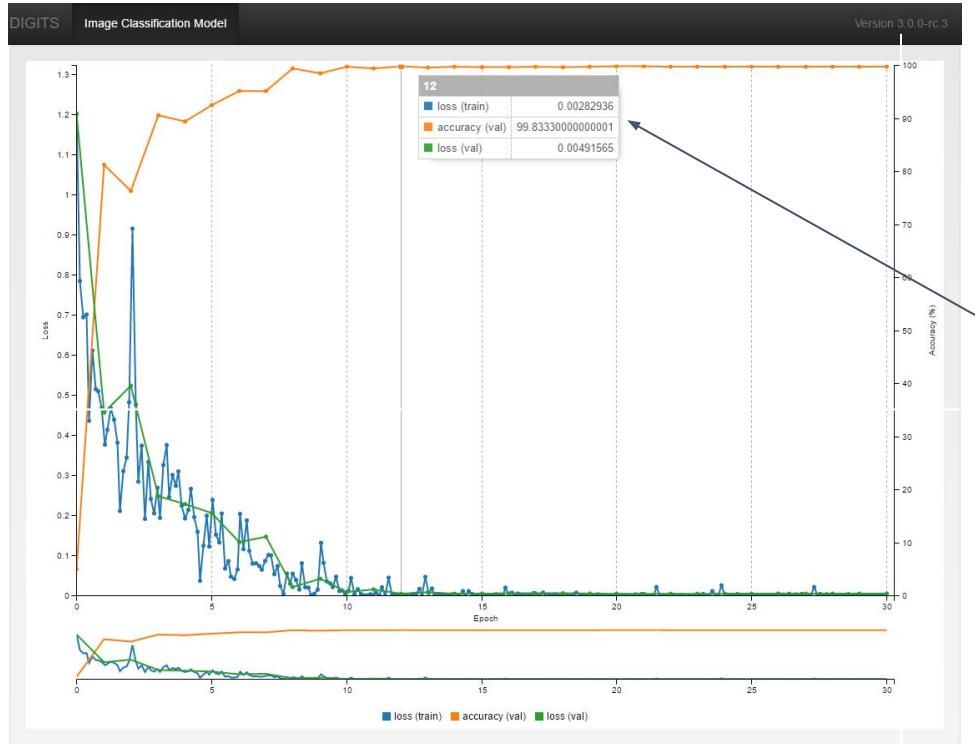
Do a small quantity of
images & classes overfit?

Load Small Training Dataset

$n = 3,465$ Image
 $K = 3$ Classes



Trained Test Model Overfits



@ 12 epochs
Accuracy (val) 99.8%
Loss (train) 0.003
Loss (val) 0.005

Model → V001.01



Non-Convergence Test

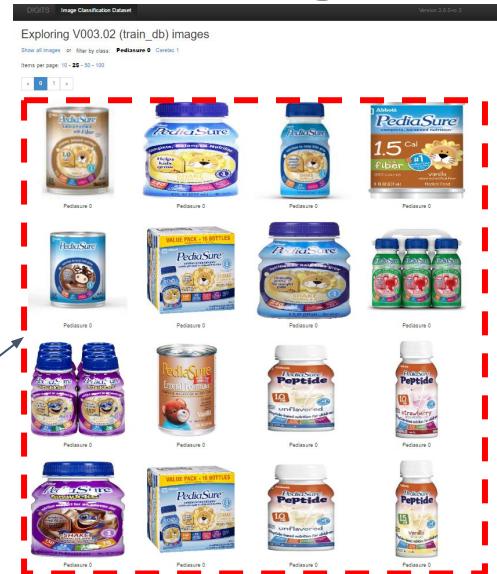
Does a small quantity of classes with highly dissimilar images converge?

Load Small Training Dataset

n = 70 Images

K = 2 Classes

Class = PediaSure
Dissimilar Images



Create DB (train)

Input File (before shuffling)

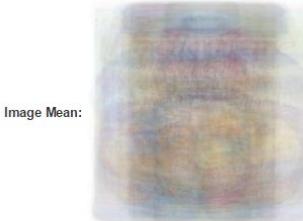
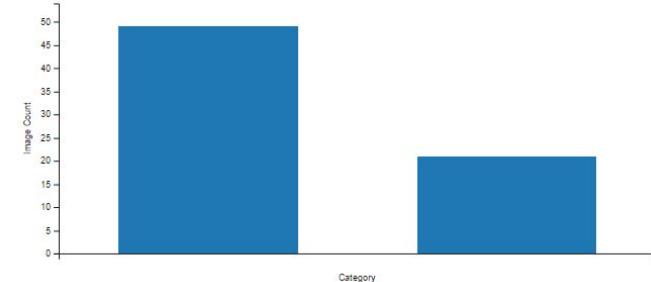
/home/billgold/SmartCart/ProtoType/v003/DIGITS_Training_Overfit.csv

DB Creation log file

create_train_db.log

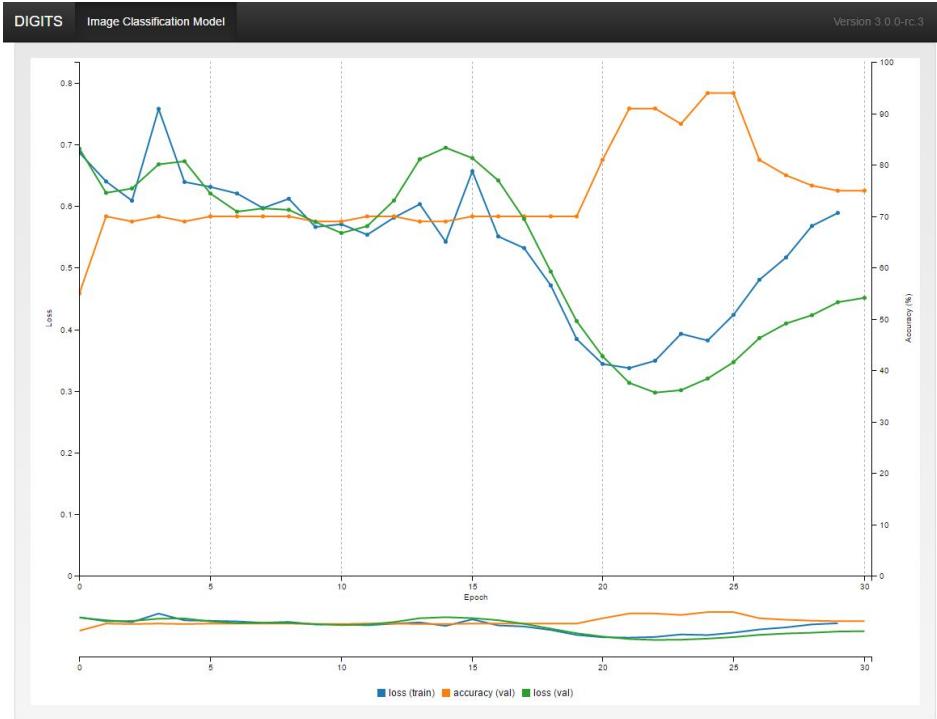
DB Entries

70



Explore the db

Model Does not Converge



Model → V001.02

v. Train to the Brand

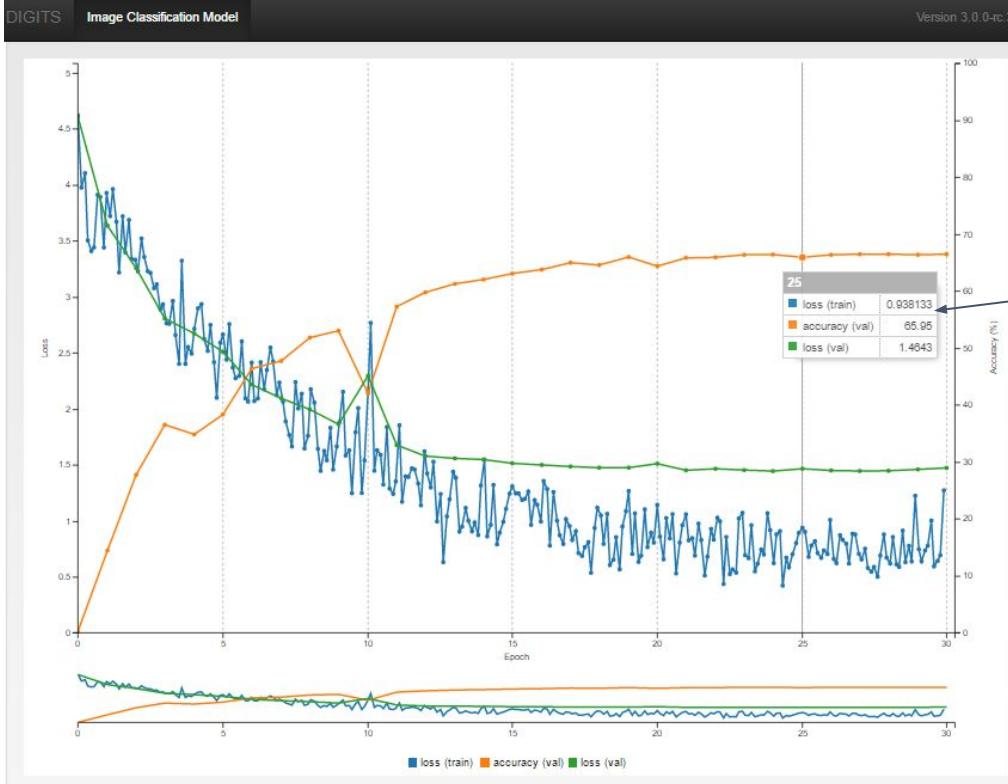
Split Data

Randomize Order of Training Data

$n = 4,552 + 1,975$ Images

$K = 99$ Classes

Training Results



@ 25 epochs
Accuracy 66%
Loss (Train) 0.94
Loss (Val) 1.5

Model → V002.01

Issue #1 - Image Exclusions Needed



0a2da1ab0a7e6452c7a6ac5efcd24
23e_000000



0a2da1ab0a7e6452c7a6ac5efcd24
23e_000001



0aa1b675f5cdde2fd0d8878a88b7a3
db1_000000



0ae4c58f2eaca46260f42512de7f08
28_000000



0ae4c58f2eaca46260f42512de7f08
28_000001

Nutrition Facts	Amount Per Serving
Serving Size 1 pack	0g Total Carb
Trans Fat 0g	Dietary Fiber 2g
Contain 2 packs	Sodium 35mg Sugars 3g
Calories 40	Potassium 140mg Protein 2g
% Daily Value (DV)	• Protein 3% • Vitamin A 25% (100% from beta carotene) • Vitamin C 0% • Calcium 2% • Iron 4%

0ae4c58f2eaca46260f42512de7f08
28_000002

INGREDIENTS: PEAS, CARROTS, WATER, SPINACH.

0ae4c58f2eaca46260f42512de7f08
28_000003



0b0309fa665e46ab3e8fcc083ef21f
23_000000



0b08505c561cf2868dbf677c8fe13
6d7_000000



0b08505c561cf2868dbf677c8fe13
6d7_000001



0b08505c561cf2868dbf677c8fe13
6d7_000002

Nutrition Facts	Amount Per Serving
Serving Size 1 Jarlet (1.25 oz)	0g Total Carb
Calories From Fat 0g	Dietary Fiber 0g
Total Fat 0g	Sodium 0mg Sugars 0g
Unsaturated Fat 0g	Cholesterol 0mg
Trans Fat 0g	Protein 0g
Carbohydrate 0g	Fat
Fat 0g	• Saturated Fat 0g
Cholesterol 0mg	• Trans Fat 0g
Protein 0g	Carbohydrate 0g
Fat	• Sugars 0g
Cholesterol	• Protein 0g
Protein	% Daily Value
	• Protein 0% • Vitamin A 0% • Vitamin C 0% • Calcium 0% • Iron 0%

0b08505c561cf2868dbf677c8fe13
6d7_000003

Effort to Cleanse Image Database

20k

Images

@15

Seconds per Image

83

Hours to Process (Yikes!!)

Solution #1

Pre-Process Images

The screenshot shows a software interface for pre-processing images. On the left, there are three separate nutrition facts labels from Gerber products. In the center, a search dialog box is open, overlaid on the labels. The dialog has a red dashed border around its top half. It contains several radio buttons for selecting product types: Single, Multiple, Nutrition Label (which is selected), Ingredients Label, Size, Other, Case, and Pack. To the right of these buttons is a dropdown menu set to "Gerber". Below the dropdown are buttons for "Previous", "Next", and "Next Blank". A search input field contains the number "8770" with a "Go" button next to it. Below the search field are several checkboxes: "Duplicate Mode", "Jump to Blank Mode", "Barcode Displayed", and "QR Code Displayed". At the bottom of the dialog are four small thumbnail images of baby cereal boxes. The entire dialog is set against a light gray background.

Add
Categorical
Variable
Image Type
Code
Code an
Image Editor

Issue #2 - Similar Image Different Truth

Image

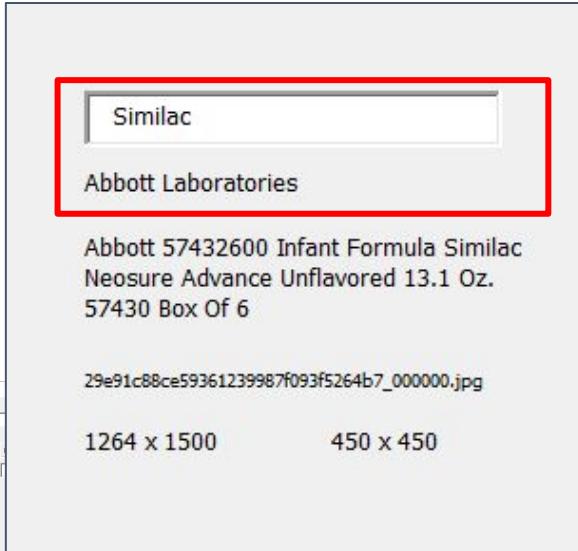


Truth
Class

Abbott Laboratories

Similac

Solution #2



Add New
Dependent
Variable
Brand
Override

Solution #2

Pre-Process Images

The interface includes the following controls and information:

- Single Product** (radio button selected)
- Multiple Products** (radio button)
- Nutrition Label** (radio button)
- Ingredients Label** (radio button)
- Size** (radio button)
- Other** (radio button)
- Case** (radio button)
- Pack** (radio button)

Search fields and buttons:
Search term: **Similac**
Page navigation: Previous, Next, Next Blank
Image index: <=, >=, 100
Image ID: 18425, Go
Mode checkboxes:

- Duplicate Mode
- Jump to Blank Mode
- Bar Code Displayed
- QR Code Displayed

Below the search area, there is a grid of thumbnail images representing processed versions of the Similac can. One specific thumbnail in the bottom row is highlighted with a red border.

Details for the highlighted image:
Caption: **NeoSure Advance Infant Formula with Iron 13.1oz**
File name: **e98052a9ca2b8af784f45bba905cd908_000000.jpg**
Dimensions: **1298 x 1500**, **450 x 450**
ID: **70074574318**

Faster Bulk Updates in Table Mode

prdurlimgraw.2016.02.18 - Excel

Bill Gold

X37	A	C	D	E	T	U	V	W	X	Z
prdID	prdURL	prdBrandRaw	prdName	imgBatchDt	imgTransCd	iRandom	imgStatus	prdBrandNameRefined		
32	a34b773a6f58866b3fdce0afc52f3	http://www.AB	Earth Mama Angel Baby Organic Herbal Utterine Tonic Tea, Thir	2016-01-01						
33	d95b8494731859c1a82020f109aad11e	http://www.AB	Plum Organics							
34	d95b8494731859c1a82020f109aad11e	http://www.AB	Plum Organics							
35	d95b8494731859c1a82020f109aad11e	http://www.AB	Plum Organics							
36	d95b8494731859c1a82020f109aad11e	http://www.AB	Plum Organics							
37	1b598bd18502b2664e96cd9b7ea4452	http://www.Abbot	Cyclinex - Am							
38	a28f16a17fdab864ebcf026a189b5c	http://www.Abbot	Ensure (Forms							
39	9788d37240e84f5e7fdaed25aa5635a	http://www.Abbot	Ensure (Forms							
40	aadd80832537029b59e101085fb5b8	http://www.Abbot	Ensure® Orig							
41	aadd80832537029b59e101085fb5b8	http://www.Abbot	Ensure® Orig							
42	49ced915f5dc233657b44b9285255b4	http://www.Abbot	Hi-Cal Oral Su							
43	4a50ce82b2a4932cbe4916e7aceb	http://www.Abbot	Ketonex							
44	f2880fc91aeb5dc0263025511b5c62d	http://www.Abbot	Pediasure (Strawberry) 8-Fl-Oz Can Ready To Use - 1 Case Of 2	2016-01-01						
45	a5fb2370743fd293ba97e936fb0923	http://www.Abbot	Pediasure (Vanilla) 24/8-Fl-Oz Can - 1 Case Of 24	2016-01-01						
46	e048a8446da58f1e43436e6478ace27	http://www.Abbot	Pediasure (Vanilla) 24/8-Fl-Oz Can - 1 Case Of 24	2016-01-01						
47	35907fa009639710cd050b9e60905ea	http://www.Abbot	Pediasure 1.5 Cal Vanilla 24/8 Fluid Ounce Cans - 1 Case Of 24	2016-01-01						
48	94461048e825106ed8b9fb03c15d537e	http://www.Abbot	Pediasure Complete Balanced Nutrition Ready To Use (Banana)	2016-01-01						
49	cde0657707eeb5812c84ffbdcc054920	http://www.Abbot	Pediasure Complete, Balanced Nutrition Ready To Use (Vanilla)	2016-01-01						
50	01c5477745343ca006f7087a5856dc4	http://www.Abbot	Pediasure Peptide 10.8g Kcal 24/8-Fl-Oz Bottle - 1 Case Of 24	2016-01-01						
51	0625632a92d5ec2765925a36b5d859a	http://www.Abbot	Pediasure Peptide Vanilla 24/8-Fl-Oz Bottle - 1 Case Of 24	2016-01-01						
52	9c0553937fbcbf01fe777544a43d62	http://www.Abbot	Pediasure Vanilla 6Pk/8-Fl-Oz Bottles - 1 Case Of 24	2016-01-01						
53	88154915704086ac1fbfa8e2747996b9	http://www.Abbot	Pediasure With Fiber Complete Balanced Nutritional Ready To U	2016-01-01						
54	3e2b32f2336da1154f599cb155a2f113	http://www.Abbot	Similac Special Care 24 With Iron And Mixed Carotenoids Ready	2016-01-01						
55	2fadbbd2b7391fa8b0ffcc834a75a07	http://www.Abbot	Suplena Vanilla Ready To Drink 24/8Oz Cans - 1 Case Of 24	2016-01-01						
56	021269a8717a65315dea218b2021ac42	http://www.Abbott	[P17] Pedia Sure Complete Chocolate Flavored Infant Food 140	2016-01-01						
57	9a934633d60220c04f0859a628804408	http://www.Abbott	[P17] Pedia Sure Complete Chocolate Flavored Infant Food 140	2016-01-01						
58	043fb8e934f1f0b6622af0814610944	http://www.Abbott	[P17] Pedia Sure Complete Chocolate Flavored Infant Food 700	2016-01-01						
59	74b4ed4c34c871fb65fd2c03fc4f86	http://www.Abbott	[P17] Pedia Sure Complete Chocolate Flavored Infant Food 700	2016-01-01						
60	c8186e7f02252104451847287755f6de	http://www.Abbott	[P17] Pedia Sure Complete Chocolate Flavored Infant Food 700	2016-01-01						
61	ee4f5ac1700fcfbfc1fe48e44e5b84	http://www.Abbott	[P17] Pedia Sure Complete Vanilla Flavored Infant Food 1200g	2016-01-01						
62	23342e550be67c67540cb3707e761135	http://www.Abbott	[P17] Pedia Sure Complete Vanilla Flavored Infant Food 1200g	2016-01-01						
63	4de3bbc10590d9840d702ff529f729	http://www.Abbott	[P17] Pedia Sure Complete Vanilla Flavored Infant Food 1200g	2016-01-01						
64	32ba8793d725b8d93fb3cf32fb14552	http://www.Abbott	[P17] Pedia Sure Complete Vanilla Flavored Infant Food 1600g	2016-01-01						
65	6ee5eaaff7100f088a8d81408d0c2b7	http://www.Abbott	[P17] Pedia Sure Complete Vanilla Flavored Infant Food 1600g	2016-01-01						

Inconsistent Original Brand

Consistent Revised Brand

Sample Code

```
Private Sub cmdNextBlank_Click()

    While Workbooks(sPrdImgXlsx).Sheets("prdurlimgraw").Range("R" & CStr(iRow)).Value <> "" And _
        Workbooks(sPrdImgXlsx).Sheets("prdurlimgraw").Range("A" & CStr(iRow)).Value <> ""
        iRow = iRow + 1
        'ImageDisplay (iRow)
    Wend
    ImageDisplay (iRow)

End Sub
```

(General) ImageDisplay

```
Sub ImageDisplay(piRow As Integer)
    Dim simgProdType As String
    Dim simgPath As String

    optSingle.Value = False
    optMultiple.Value = False
    optIngrediants.Value = False
    optNutrition.Value = False
    optSize.Value = False
    optOther.Value = False
    optCase.Value = False
    optPack.Value = False

    simgProdType = Workbooks(sPrdImgXlsx).Sheets("prdurlimgraw").Range("R" & CStr(piRow)).Value

    If simgProdType = "SING" Then optSingle.Value = True
    If simgProdType = "MULT" Then optMultiple.Value = True
    If simgProdType = "ING" Then optIngrediants.Value = True
    If simgProdType = "NUT" Then optNutrition.Value = True
    If simgProdType = "SIZE" Then optSize.Value = True
    If simgProdType = "OTH" Then optOther.Value = True
    If simgProdType = "CASE" Then optCase.Value = True
    If simgProdType = "PACK" Then optPack.Value = True

    If Workbooks(sPrdImgXlsx).Sheets("prdurlimgraw").Range("P" & CStr(piRow)).Value <> "" Then
        cbBarCode.Value = True
    Else
        cbBarCode.Value = False
    End If
End Sub
```

Issue #3

1

Classes with Insufficient Images

2

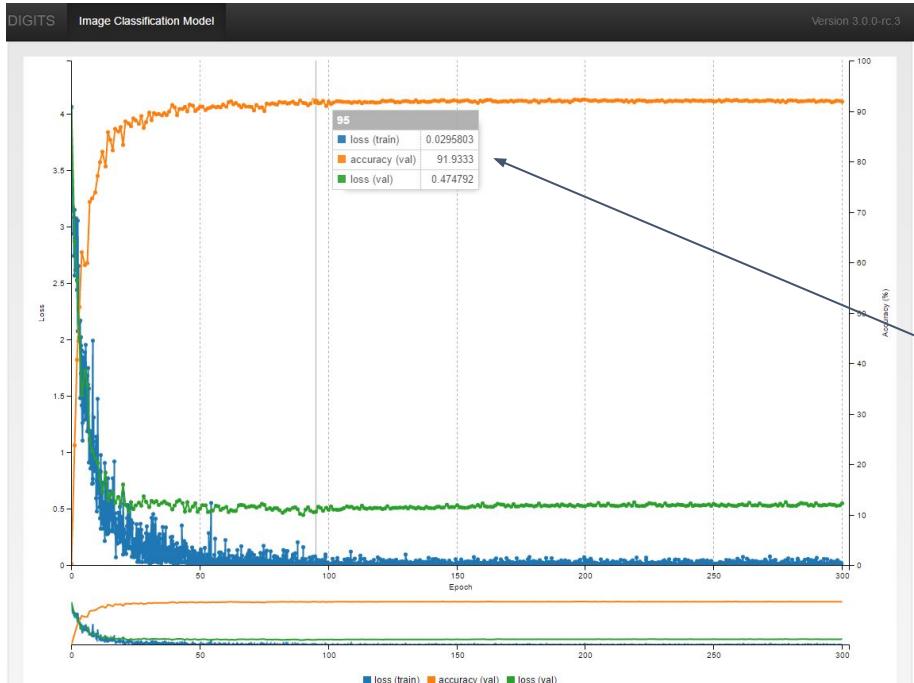
Solution

```
48 q <- paste("SELECT
49   "
50   "
51   "FROM
52   "WHERE
53   " AND
54   " AND
55   " AND
56   "GROUP BY
57   "HAVING
58   sep='')
59 dbGetQuery(conn, q)
```

pir.prdBrandNameRefined, ",
count(DISTINCT pir.prdID) as prdIDCnt, ",
count (DISTINCT pir.imgNameRefined) as imgNameCnt ",
PrdRaw pr, PrdImgRaw pir, BrandsRaw br ",
pr.prdID = pir.prdID ",
pir.prdBrandNameRaw = br.BrandNameRaw ",
br.TypeCd != 'RET' ",
pir.imgTypeCd = 'SING' ",
pir.prdBrandNameRefined ",
count(DISTINCT pir.prdID) >= 20 ",

Brand Name (Raw)	Product Count	Image Count
360 Skin Care	1	4
AAA	1	6
AB	4	4
ABBOTT NUTRITION	1	2
ALTEZA	1	1
Abbot	16	16
Abbott	38	39
Abbott Laboratories	1	1
Abbott Nutrition	200	240
Abbott Nutrition (Ross)	25	26
Abbott Nutritional	2	2
Abercrombie & Fitch	1	1
Aikureo	2	2
Alacta	1	1
Alfamino Infant	1	1
AliMed	5	5
Allegro	1	1
Almo Nature	1	1
Alpen	1	1
Amara	1	1
Amara Baby Food	2	7
Amarelli	1	3
Among Friends Baking Mixes	1	1
Amul	1	1
Andalou Naturals	1	1
Anna's	1	2
Annabel Karmel	18	18
Annie's Homegrown	2	2
Apple & Eve	2	4

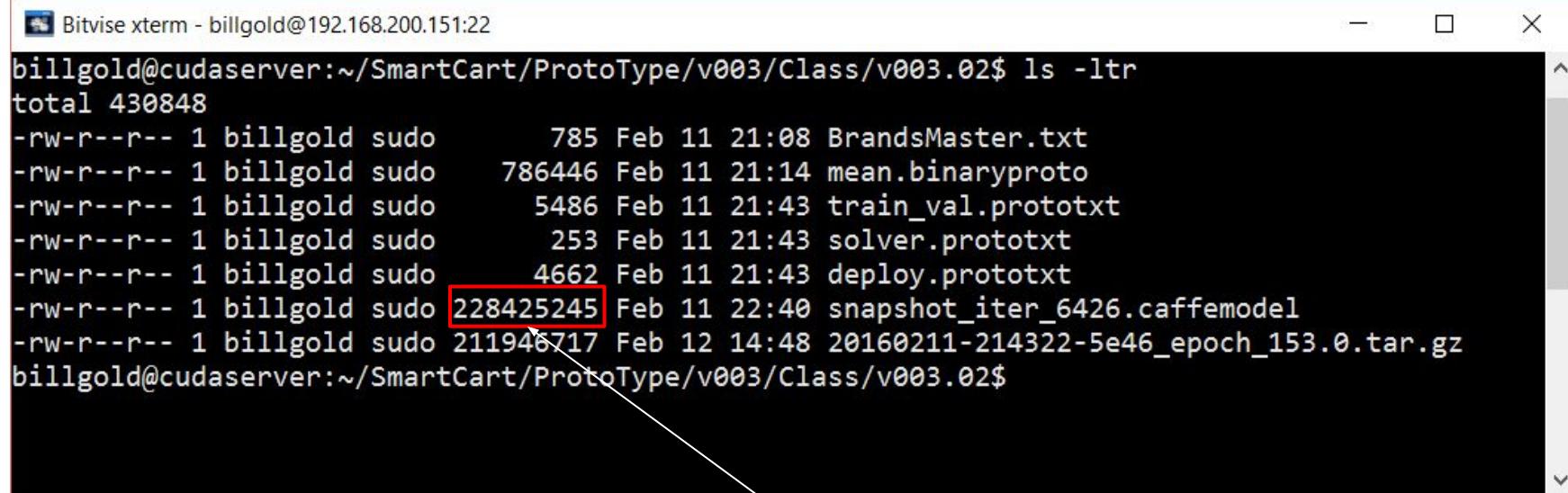
Training Results



@ 94 epochs
Accuracy (val) 92%
Loss (train) 0.03
Loss (val) 0.47

Model → V003.02

Issue #4



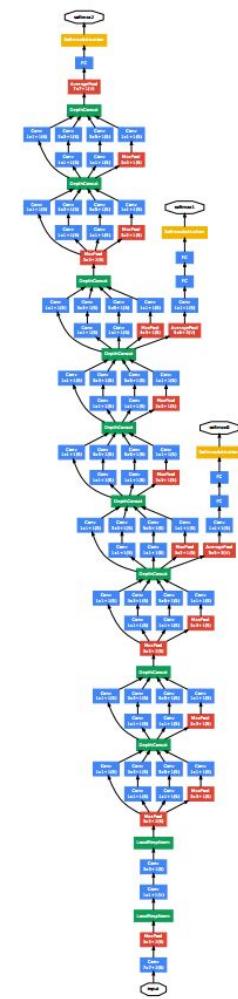
```
Bitvise xterm - billgold@192.168.200.151:22
billgold@cudaserver:~/SmartCart/ProtoType/v003/Class/v003.02$ ls -ltr
total 430848
-rw-r--r-- 1 billgold sudo      785 Feb 11 21:08 BrandsMaster.txt
-rw-r--r-- 1 billgold sudo  786446 Feb 11 21:14 mean.binaryproto
-rw-r--r-- 1 billgold sudo   5486 Feb 11 21:43 train_val.prototxt
-rw-r--r-- 1 billgold sudo    253 Feb 11 21:43 solver.prototxt
-rw-r--r-- 1 billgold sudo   4662 Feb 11 21:43 deploy.prototxt
-rw-r--r-- 1 billgold sudo 228425245 Feb 11 22:40 snapshot_iter_6426.caffemodel
-rw-r--r-- 1 billgold sudo 211946717 Feb 12 14:48 20160211-214322-5e46_epoch_153.0.tar.gz
billgold@cudaserver:~/SmartCart/ProtoType/v003/Class/v003.02$
```

Deployed Model > 200MB

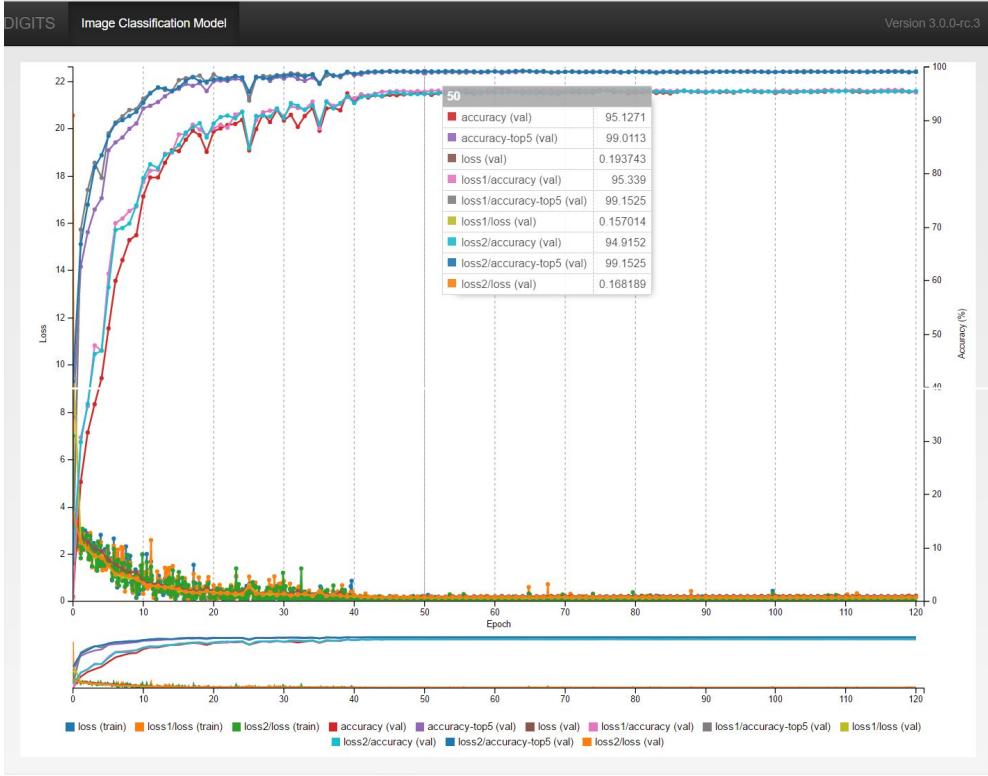
Solution #4

Use GoogLeNet¹ for CNN Architecture

- 1 - Eliminate FC layer (smaller deployment size)
- 2 - Deeper Network (higher accuracy)



GoogLe Results



@ 50 Epochs
95% Accuracy

35MB Model

Model → V003.07

Select Weights & Activations

Test Image



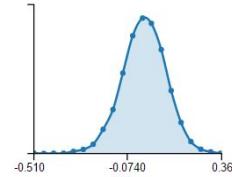
Predictions

Beech-Nut 0	100.0%
Little Duck 18	0.0%
Earth's Best 3	0.0%
Hot Kid 16	0.0%
Gerber 9	0.0%

conv1/7x7_s2

Weights (Convolution layer)
9,472 learned parameters

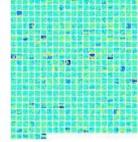
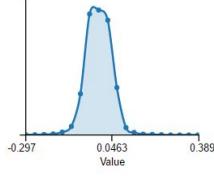
Data shape: (64, 3, 7, 7)
Mean: -0.000147946
Std deviation: 0.0980401



inception_3a/5x5

Weights (Convolution layer)
12,832 learned parameters

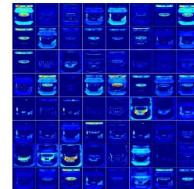
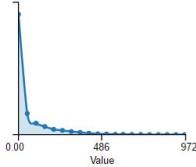
Data shape: (32, 16, 5, 5)
Mean: -0.00532596
Std deviation: 0.0478775



conv1/7x7_s2

Activation

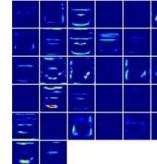
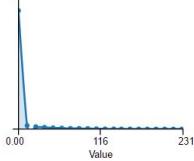
Data shape: (64, 112, 112)
Mean: 60.8099
Std deviation: 105.639



inception_3a/5x5

Activation

Data shape: (32, 28, 28)
Mean: 4.66871
Std deviation: 17.5205



F. Deployment

71

7. Next Steps

- Complete deployment to mobile device
- Review Filters
- Product as the dependent variable
 - Experiment with Hierarchical Softmax ¹ & OCR
- Integrate In Store GPS data
- Image Filters are attributed to 13% of GoogLeNet Errors ²
(2nd largest source)
- Design network architecture

1 - Distributed Representations of Words and Phrases and their Compositionality - <http://arxiv.org/pdf/1310.4546.pdf>

2 - Andrej Karpathy blog - <http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/>

Today's Cart Is Largely Unchanged



circa 1940



circa 2010