Constructing Caveat Contracts from API Documentation and its Applications

Thien Bui-Nguyen
Dr. Zhenchang Xing
5/11/19

Introduction

How accessible are reference documentation for APIs?

What if we can link API documents to code?

Application Programming Interfaces (APIs)

Example from Facebook SDK for Javascript Reference document

Core Methods

Method	Description
`.init()`	Used to initialize and setup the SDK. All other SDK methods must be called after this one.
`.api()`	Make an API call to the Graph API.
`.ui()`	Used to trigger different forms of Facebook created UI dialogs, such as the Feed dialog, or the Requests dialog.

Constraints related to some API component

API Caveat Example #1 - Java API (ArrayList class)

get

```
public E get(int index)
```

Returns the element at the specified position in this list.

Specified by:

`get` in interface `List<E>`

Specified by:

`get` in class `AbstractList<E>`

Parameters:

`index` - index of the element to return

Returns:

the element at the specified position in this list

Throws:

`IndexOutOfBoundsException` - if the index is out of range (`index < 0 || index >= size()`)

API Caveat Example #2 - Android API (SpringAnimation class)

skipToEnd

added in version 25.4.0

```
void skipToEnd ()
```

Skips to the end of the animation. If the spring is undamped, an `IllegalStateException` will be thrown, as the animation would never reach to an end. It is recommended to check `canSkipToEnd()` before calling this method. This method should only be called on main thread. If animation is not running, no-op.

Throws	
`IllegalStateException`	if the spring is undamped (i.e. damping ratio = 0)
`AndroidRuntimeException`	if this method is not called on the main thread

The Problem

“You don’t know what you don’t know”

Locating API Misuse Examples in GitHub Data

- Previous work linked code examples from Stack Overflow
- Does the same approach work in a different domain?
- Found negative results
- Explicit caveats better suited for real-time applications

Constructing Caveat Contracts from API Documentation

Main Idea:

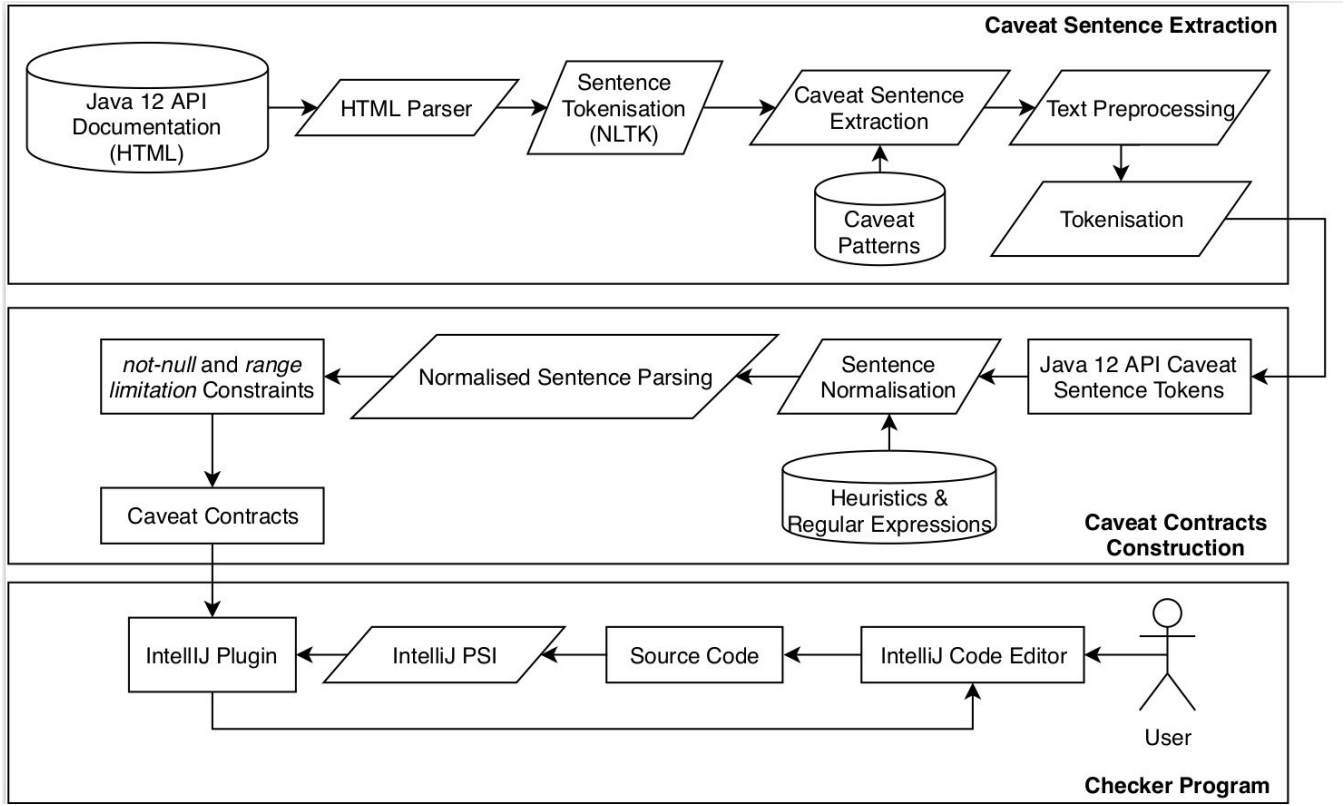
1. *Exception caveats* explicitly describe conditions for exceptions
2. Can map these caveats to code contracts

Code Contracts

Specify requirements for code

E.g. Method M requires parameter P to be positive

Design Architecture



Constructing Caveat Contracts

- Sentence normalisation
  - E.g. substitute “a < 0 || b < 0” with “EXPR0”
- Assume common sentence structures
  - Subject-object-verb (English)
  - Greedy approach to extract constraints

Constructing Caveat Contracts - Toy Example

Sentence Normalisation

1. “IllegalArgumentException - if capacity is less than c.size(), or less than 1”
2. “if capacity is less than c.size(), or less than 1”
3. “if @PARAM1 is @EXPR1, or @EXPR2”

Caveat Contracts Construction

4. Analyse tokens/words from left to right
5. Obtain constraints: “capacity < c.size()” and “capacity < 1”
6. Represent as code contracts

Caveat Contract Example

JSON representation

```
{
        "className":"java.util.concurrent.ArrayBlockingQueue",
        “api":"ArrayBlockingQueue",
         "signature":"public ArrayBlockingQueue(int capacity, boolean fair,
                              Collection<?extends E> c)",
        "rangeRules":[
                { "Param":0, "op":"<", "Constraint":"1"},
                { "Param":0, "op":"<", "Constraint":"c.size()"}
        ]
}
```

Caveat Contracts Results

Extracted 4,694 unique caveat constraints
(Java 12 API Documentation)

Evaluated approach (random sampling, manual labelling):

Accuracy	Precision	Recall	F1 Score
0.77	0.96	0.73	0.83

Static Code Analysis

Examining of code without executing the program

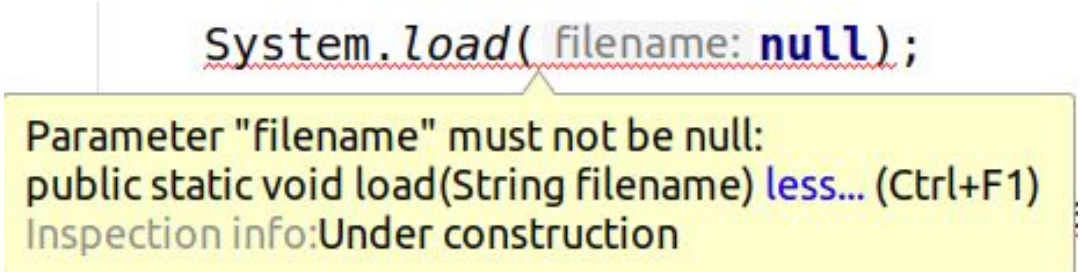
Results: Plugin Off

```
public static void main(String[] args) {
    // Not null constraints violated
    SchemaFactory schemaFactory = SchemaFactory.newInstance(null);

    Font font = new Font( name: "TimesRoman", Font.PLAIN,  size: 12);
    font = Font.getFont( nm: null,  font: null);

    System.load( filename: null);

    // Range limitation constraints violated
    BasicStroke basicStroke = new BasicStroke( width: -1);

    Random r = new Random();
    r.nextInt( bound: -1);

    MessageInfo messageInfo1 = MessageInfo.createOutgoing( address: null,  streamNumber: -1);
    MessageInfo messageInfo2 = MessageInfo.createOutgoing( address: null,  streamNumber: 65537);
}
```

Results: Plugin On

```
public static void main(String[] args) {
    // Not null constraints violated
    SchemaFactory schemaFactory = SchemaFactory.newInstance(null);

    Font font;
    font = Font.getFont( nm: null,  font: null);

    System.load( filename: null);

    // Range limitation constraints violated
    BasicStroke basicStroke = new BasicStroke( width: -1);

    Random r = new Random();
    r.nextInt( bound: -1);

    MessageInfo messageInfo1 = MessageInfo.createOutgoing( address: null,  streamNumber: -1);
    MessageInfo messageInfo2 = MessageInfo.createOutgoing( address: null,  streamNumber: 65537);
}
```

Results: Plugin On (Error Message Example)



Summary

- Linking API documentation to code
- Attempted previous solutions on GitHub data
  - Poor results found
  - Discovered lexical gap between GitHub and API document text
- Looked at constructing caveat contracts
  - Proposed simple parsing technique
- Developed proof-of-concept checker plugin

Australian National University

Design Compsition Critique

Presenter: Sidong Feng (u6063820)

Supervisors: Dr Zhenchang Xing

Dr Chunyang Chen

Background

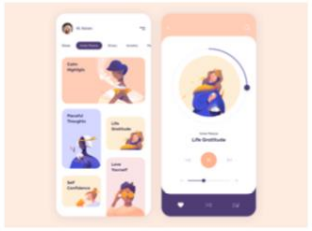
Graphic User Interface (GUI) design online sharing is a routine activity among designers

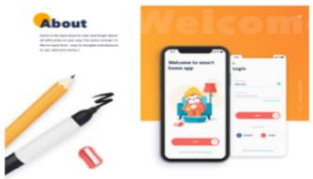
- boost portfolio
- gain inspiration
- understand trend

dribbble

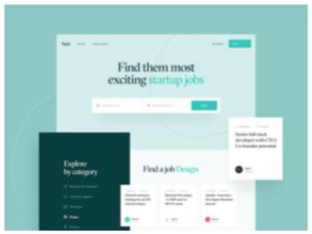
GRAPHIC BURGER

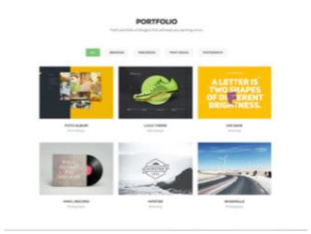
Mobile

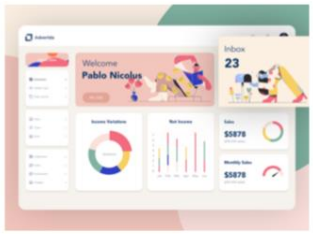




Website







Dribbble GraphicBurger Behance

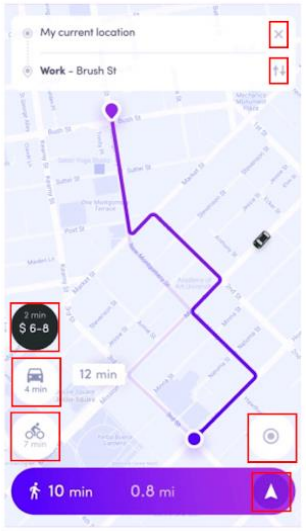
Motivation

1. Inefficient tagging-based search due to **unclear**, **incomplete** and **inconsistent** tags annotated by uploader.
    - Search for **e-commerce** cannot retrieve any result of **ecommerce**, vice versa.

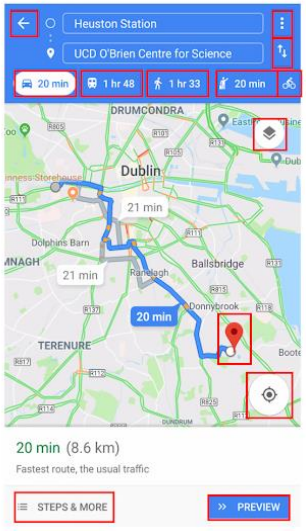
Motivation

2. Design Practicality
   - Creative design artworks only
   - **BUT** practical GUI designs in real applications is needed

Artwork

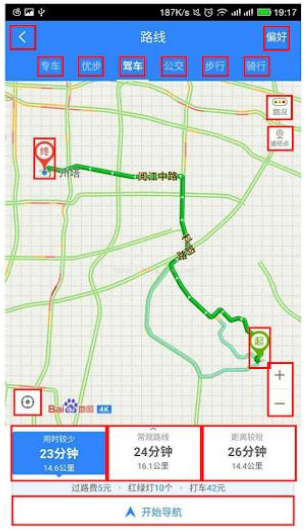


Google Map



HERE WeGo

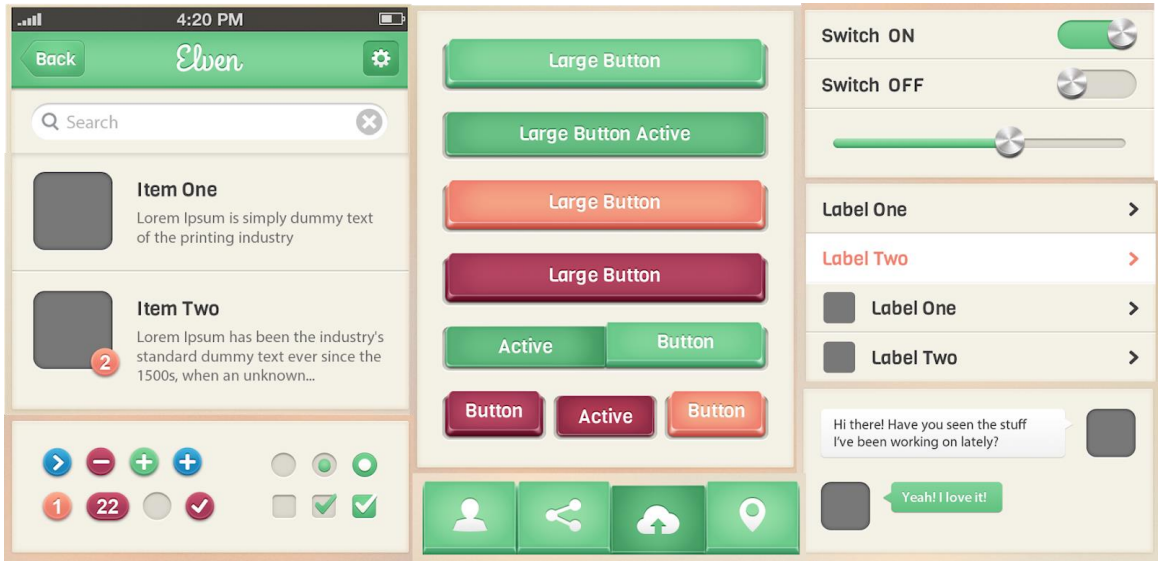
Baidu Map



Tencent Map

Motivation

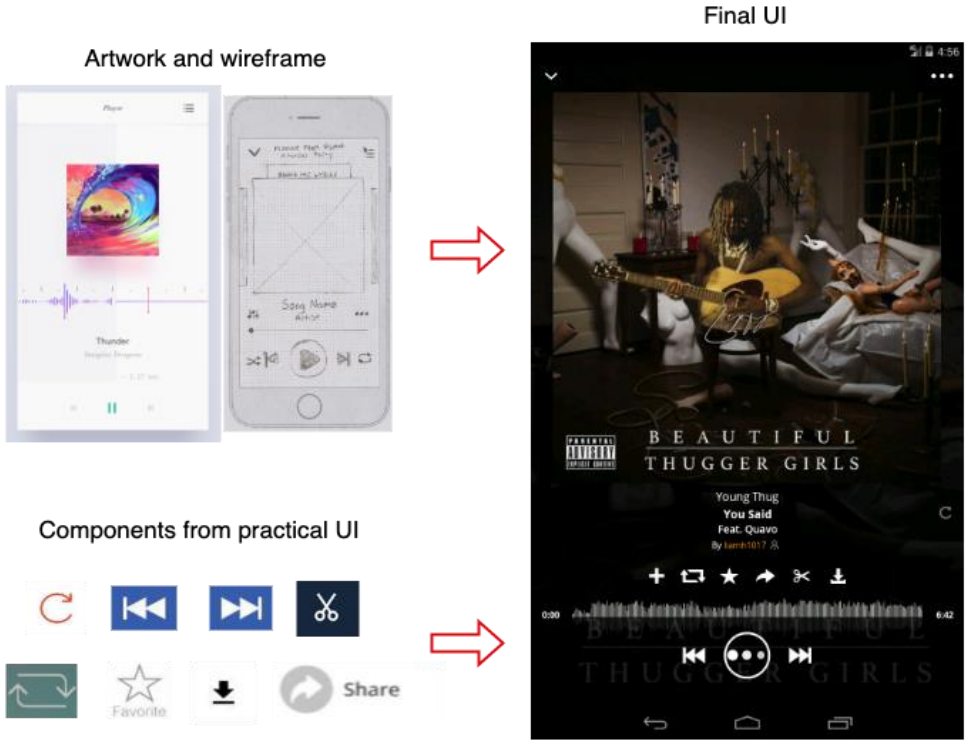
3. Design Granularity
   - Whole design of the GUI only
   - **BUT** the detailed design of the GUI components is needed



Objective

Design Gallery (http://mui-collection.herokuapp.com/)

- Support advanced tagging-based search for whole UI artwork design
- Support multi-faceted search for component-oriented practical design



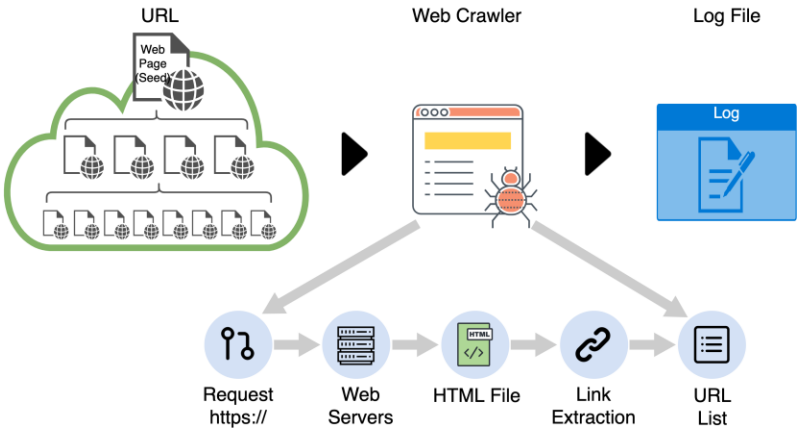
Whole UI Artwork Search

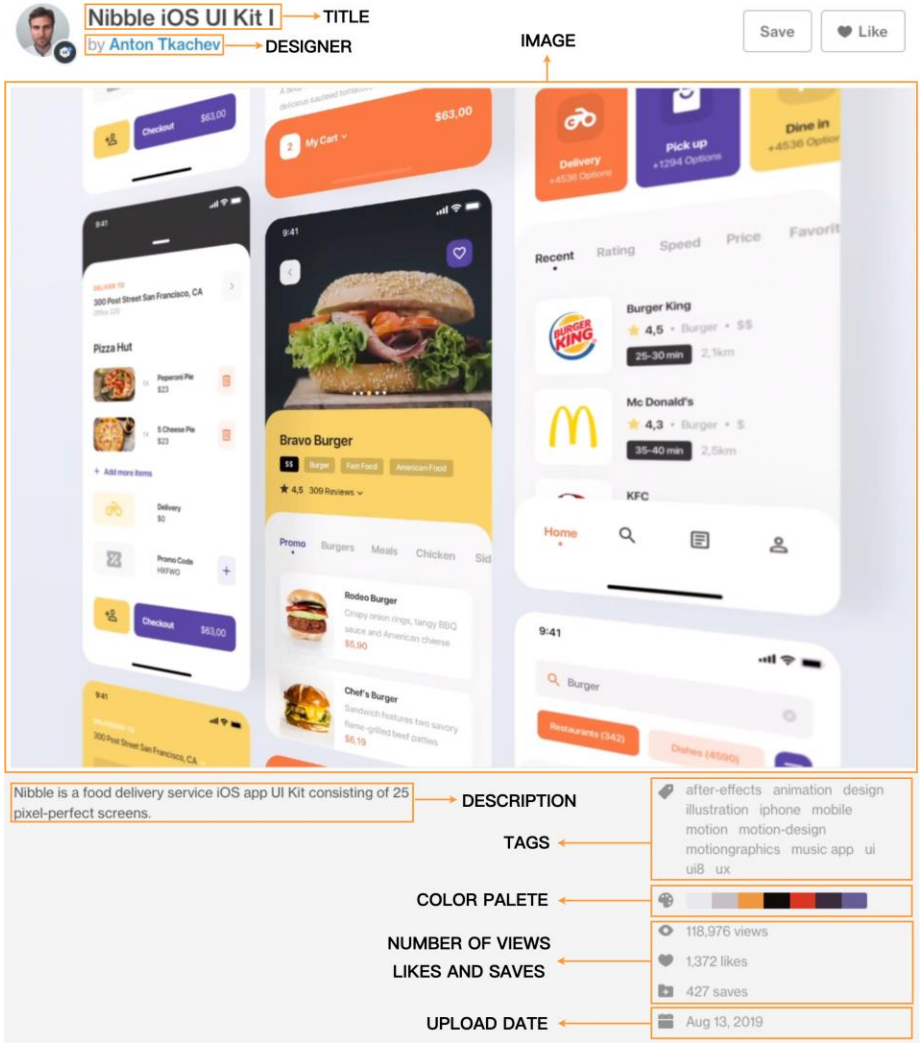
- Problem Definition
  - undescriptive tags => unclear
  - same-meaning tags => inconsistent
  - missing tags => incomplete

- Approach
  - A structured UI semantic vocabulary (solve unclear and inconsistent)
  - A tag prediction model (solve incomplete)

Data Collection

- Breadth-First method[1] to crawl 61,700 UI designs from Dribbble [2] and its associated information

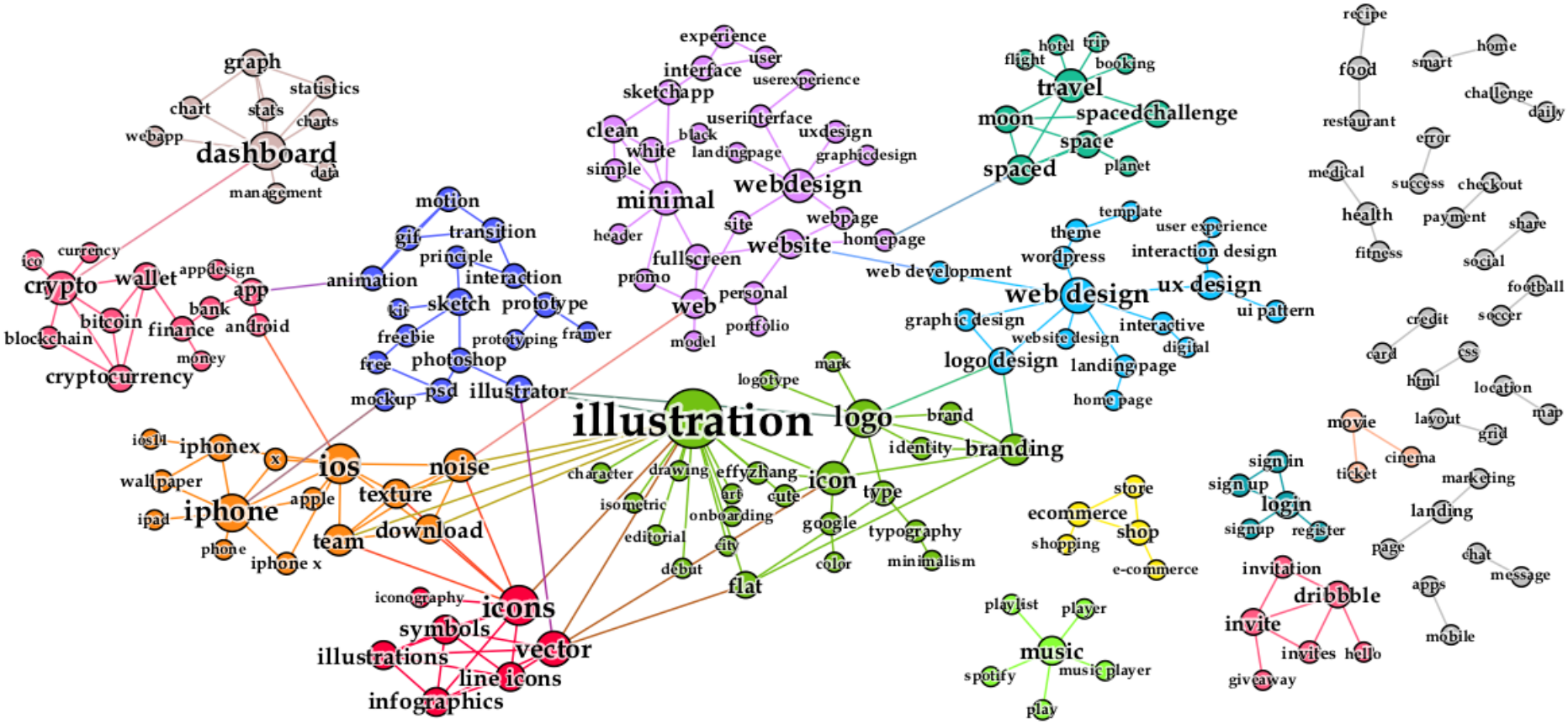




UI Semantics

Problem	Example	Definition	Solution
Implicit Correlation	(payment, checkout)	Association Rule Mining [3]	Aprori algorithm [4]
Category Classification	Sport: gym, workout	Consensus-driven approach	Google's Material Design
Synonyms and misspelling	visualisation, visualizatio n, vizualization	Semi-automatic method [5]	Word embedding model NLTK [6]
Abbreviation	ui, **u**ser **i**nterface	Rule based method	character is in the same order, first character, etc.

UI Semantics



CATEGORY	ASSOCIATED TAG NAME
PLATFORM	
Website:	Website, Web, Mac, Macbook
Mobile:	Mobile, Phone, IOS, Iphone, Android
Tablet:	Tablet, Ipad, Ipadpro
COLOR	
White:	White
Yellow:	Yellow, Golden, Orange
Red:	Red
Pink:	Pink
Purple:	Purple
Blue:	Blue, DarkBlue, SkyBlue
Green:	Green, DarkGreen, Aquamarine
Grey:	Grey, Silver, DarkGray
Brown:	Brown
Black:	Black
APP FUNCTIONALITY	
Music:	Music, Musicplayer, MusicApp
Food & Drink:	Food, Restaurant, Grocery, Drink
Game:	Game, Videogame
Health & Fitness:	Fitness, Health
News:	News
Sport:	Sport, Gym, Workout
E-commerce:	E-commerce, Store, OnlineShop
Social Networking:	SocialNetwork, Blog, Messenger, Facebook, Instagram, Dating, Chat
Travel:	Travel, Trip, Tourism
Weather:	WeatherApp, Temperature
Lifestyle:	Fashion, Furniture, Real Estate
Education:	Education, E-learning
Reference:	Dictionary, Atlas, Encyclopedia
Entertainment:	Movie, TV, Netflix, Youtube
Medical:	Medical, Healthcare, Hospital
Books:	DigitalReading, DigitalBookstroe
Kids:	Kids, Children
Finance:	Finance, Wallet, Bank, Business, Insurance, Marketing
Utilities:	Calculator, Clock, Measurement, WebBrowser
Navigation:	DrivingAssistance, TopographicalMaps, PublicTransitMaps
SCREEN FUNCTIONALITY	
Landing Page:	LandingPage
Login:	Login, Signin
Signup:	Signup, Registration
Checkout:	Checkout, Payment
Search:	Search
Profile:	Profile
Contact Page:	Contact, ContactPage
SCREEN LAYOUT	
Dashboard:	Dashboard
Form:	Form
Table:	Table
List:	List
Grid:	Grid
Gallery:	Gallery
Toolbar:	Toolbar, Toolbox
Chart:	Chart

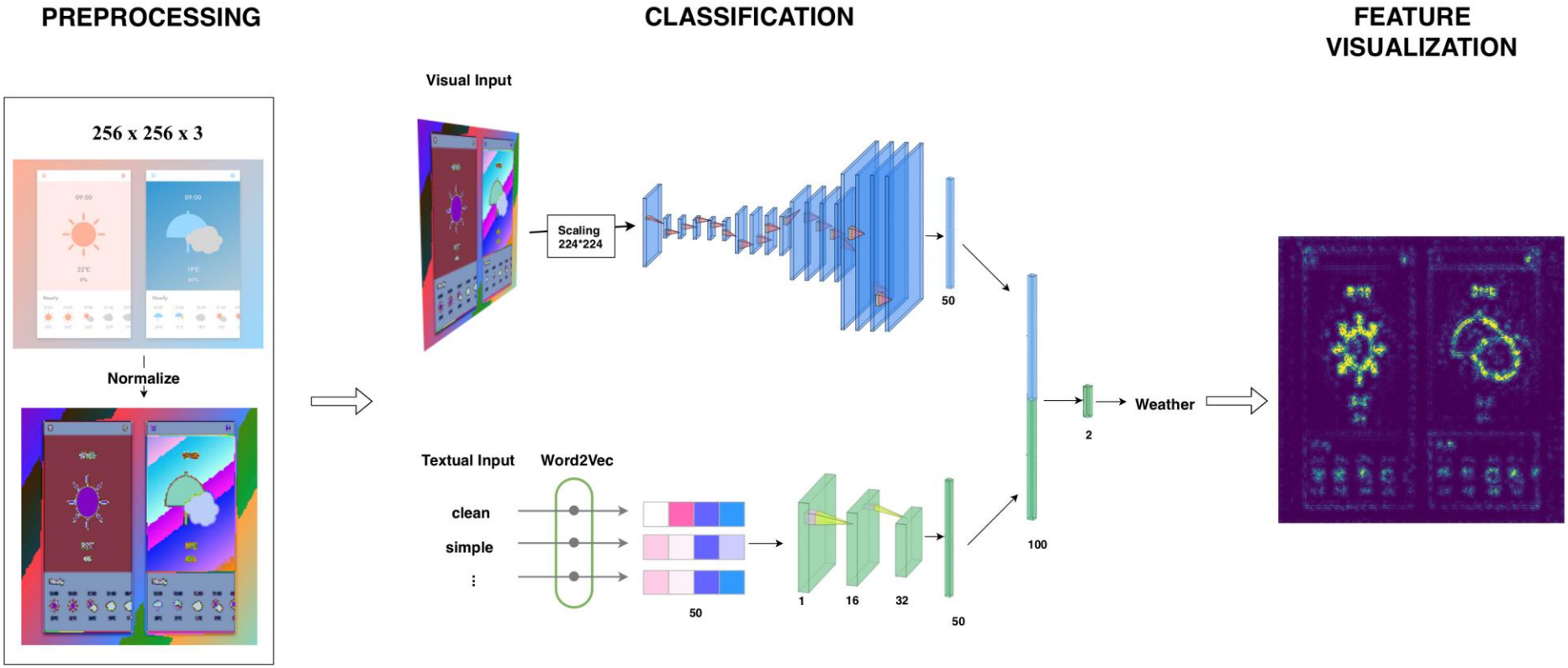
STANDARD(#)	ABBREVIATION & SYNONYMS
ui (61309):	*user interface, user_interface, user-interface design, uidesign*
website (28009):	*web, websites, webpage, website development, web design*
ux (24209):	*user experience, uxdesign, ux_design*
mobile (8554):	*mobiledesign, mobile phone, mobile_design, smartphone*
illustration (7159):	*illustation, digital_illustration, kids_illustration*
app (5887):	*apps, application, app development, app design*
landing page (5536):	*landing-page, landingpage, landing page design*
minimal (4938):	*minimalism, minimalistic, minimalist*
ios (4741):	*ios8, ios9, ios11, ios_design*
iphone (4736):	*iphone x, iphonex, iphone 7, iphone_8*
icon (4230):	*icons, icon design, icon pack*
logo (3704):	*logo design, logos, logotype*
food (2881):	*fastfood, food_blog, junk_food, doughnut*
clean (2723):	*clear, clean_design*
flat (2481):	*flat design, flat-design, flat-art*
interaction (2402):	*interactive, microinteraction, interaction design, user interaction*
dashboard (2141):	*dashboard design, dashboards*
branding (2071):	*branding design, rebranding, selfbranding*
sketch (2060):	*sketching, sketches, adobe_sketch*
ecommerce (1974):	*e-commerce, online commerce, shopping*
vector (1940):	*vectors, vector art*
product (1841):	*products, product page, product detail*
typography (1820):	*interface typography, 3d_typography*
gradient (1671):	*gradients, gradient design, blue gradient*
gif (1441):	*gifs, looping_gif*
layout (1400):	*layout design, layouts*
concept (1378):	*conceptual, concepts, concept art*
motion (1361):	*motion graphics, motion design*
responsive (1347):	*responsive design, response*
music (1251):	*music player, musician, musical, concert*
restaurant (1221):	*restaurants*
profile (1204):	*profiles, user_profile, userprofile*
travel (1197):	*travelling, travel agency, travel guide*
animation (1194):	*animations, 3danimation, 2danimation*
simple (1108):	*simply, simplistic, simple_design*
graphic (1047):	*graphics, graphic design, graphicdesigner*
color (1000):	*colors, colorful*
white (988):	*whitelabel, white design, white theme*
login (919):	*log_in, sign_in, login screen*
modern (915):	*modernistic, fashionable*

Whole UI Artwork Search

- Approach
  - ~~A structured UI semantic vocabulary (solve unclear and inconsistent)~~
  - **A tag prediction model (solve incomplete)**

Tag Prediction

- Multi-Labels problem => Multiple Binary Classifier
- Capture visual and textual information



Tag Prediction

- Intensity of Image and Tag

Image vectors	Tag vectors	Accuracy
10	90	0.5972
20	80	0.7056
30	70	0.6528
40	60	0.8134
50	50	**0.8272**
60	40	0.8222
70	30	0.8194
80	20	0.7981
90	10	0.7715
32	32	0.8008
64	64	0.8130

Tag Prediction

- Baseline: Histo+SVM, Histo+DT
- Our model achieve 82.72% accuracy

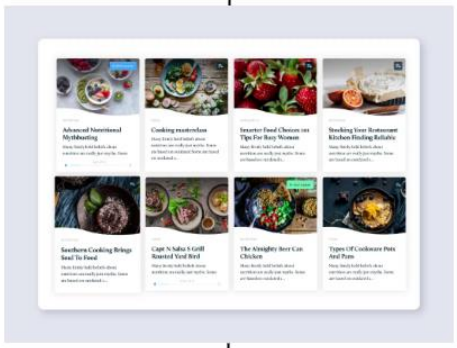
CLASSES	ACCURACY					
	Histo +SVM	Histo +DT	ResNet -pretraind	ResNet +pretraind	Tag_CNN only	ResNet +CNN +pretrained
App Function						
music	0.6636	0.5545	0.6727	0.7909	0.8545	**0.8909**
food&drink	0.5765	0.6294	0.7529	0.7882	0.7706	**0.8294**
ecommerce	0.5565	0.5726	0.6895	0.7460	0.8306	**0.8710**
finance	0.5655	0.5833	0.6964	0.7500	0.8274	**0.8512**
travel	0.5211	0.5842	0.7316	0.7053	0.8053	**0.8474**
game	0.5814	0.5814	0.8062	0.7984	0.7597	**0.8605**
weather	0.5745	0.7021	0.7447	0.7872	0.8085	**0.8298**
sport	0.4220	0.6147	0.6147	0.6239	0.7064	**0.7798**
Color						
yellow	0.5865	**0.7596**	0.7404	0.7404	0.6442	0.7500
red	0.6667	0.7083	0.8194	**0.8472**	0.6111	**0.8472**
pink	0.7609	0.6522	0.7826	0.7391	0.6522	**0.8261**
blue	0.6600	0.6800	0.7700	0.7400	0.6800	**0.8700**
green	0.7000	**0.8714**	0.8286	0.7714	0.6571	0.7857
white	0.6111	0.6111	0.7778	0.7333	0.7333	**0.7888**
black	0.6241	0.6015	0.8496	0.8271	0.6617	**0.8571**
Screen Function						
landing page	0.5465	0.5346	0.7106	0.7017	0.7947	**0.8115**
signup	0.4907	0.5556	0.7731	0.7130	0.7361	**0.7778**
checkout	0.5545	0.4182	0.6000	0.7091	0.7545	**0.8000**
profile	0.4667	0.5538	0.5487	0.6513	**0.9026**	0.7590
Screen Layout						
dashboard	0.5867	0.6067	0.7600	0.7933	0.7867	**0.8800**
chart	0.6061	0.6667	0.7121	0.7424	**0.8485**	0.8030
form	0.5429	0.5000	0.6857	0.7429	0.5714	**0.7714**
list	0.6136	0.5909	0.7045	**0.9091**	0.6364	0.8182
grid	0.5000	0.5811	0.6351	0.6486	0.7162	**0.7432**
Platform						
mobile	0.7785	0.7847	0.8356	0.7954	0.9250	**0.9402**
website	0.7513	0.7481	0.8418	0.8224	0.8837	**0.9171**
Average	0.5965	0.6249	0.7342	0.7545	0.7522	**0.8272**

Tag Prediction

- Model Behavior

Original Tags:

7ninjas, animation, app, article, browsing, cards, catalog, design, flat, interaction, list, motion, principle, ui, website, wishlist

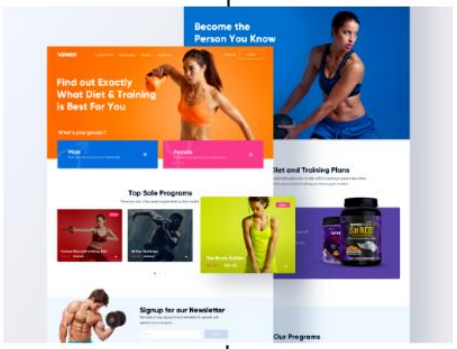


Predicted Additional Tags:

grid, food&drink

Original Tags:

colorful, colors, fitness, girl, layout, marketing website, product design, ui, ui design, ux, ux design, web, webdesign, web design



Predicted Additional Tags:

sport, yellow, blue, website, landing page

Original Tags:

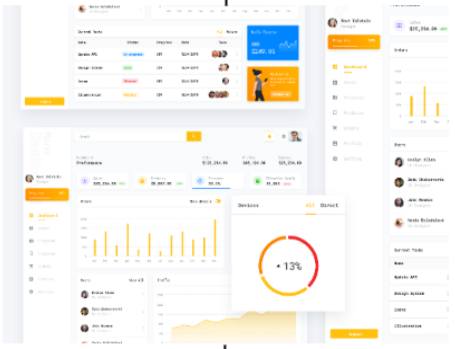
bank, chart, dark, graph, list, online paying, payment, payment history, ui, ux

Predicted Additional Tags:

finance, black, checkout, profile, mobile

Original Tags:

analytics, analytics chart, dashboard app, design, form, form elements, freebies, graph, icons, ui, uidesign, uikits, web, web design

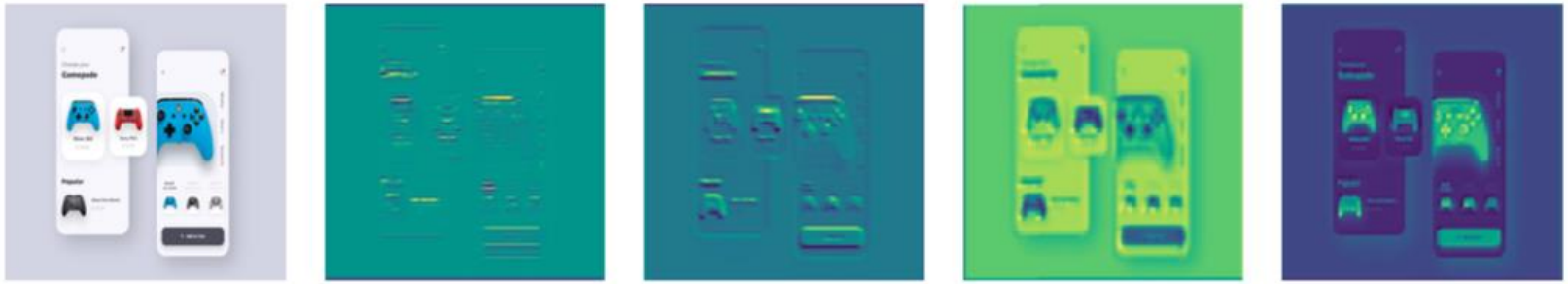


Predicted Additional Tags:

chart, yellow, finance, website, dashboard, profile

Tag Prediction

- Feature Visualization

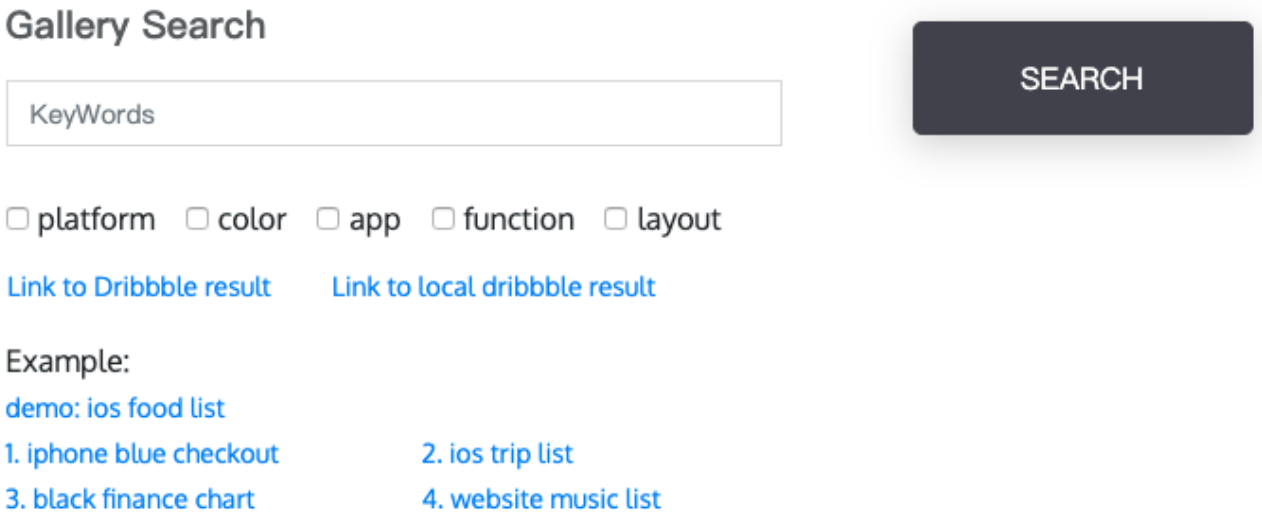


(a) food&drink (b) travel

(c) music (d) chart

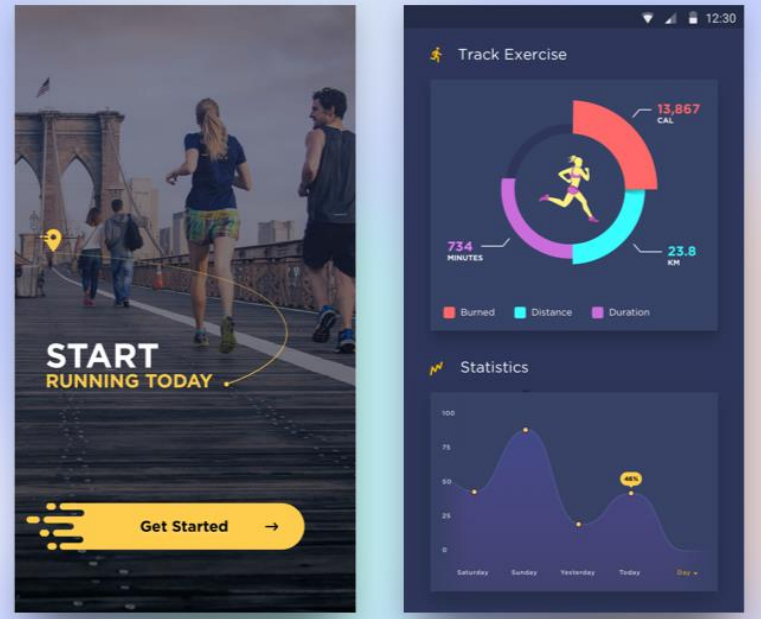
Design Gallery

- New database (16,546 UI designs)
- Advanced tagging-based search
- Existing platforms comparison
- Ranking system



Informal Feedback from Designers

- Visualizing Vocabulary



android, ios, apps, running, sport, trainer, exercise, dashbroad, creative, design, flat design, google, material, sketch design

mobile, sport, dashboard, creative, flat, google, material, design, sketch design

mobile, sport, dashboard, creative, flat, sketch design, health&fitness, login, chart, black, blue

Informal Feedback from Designers

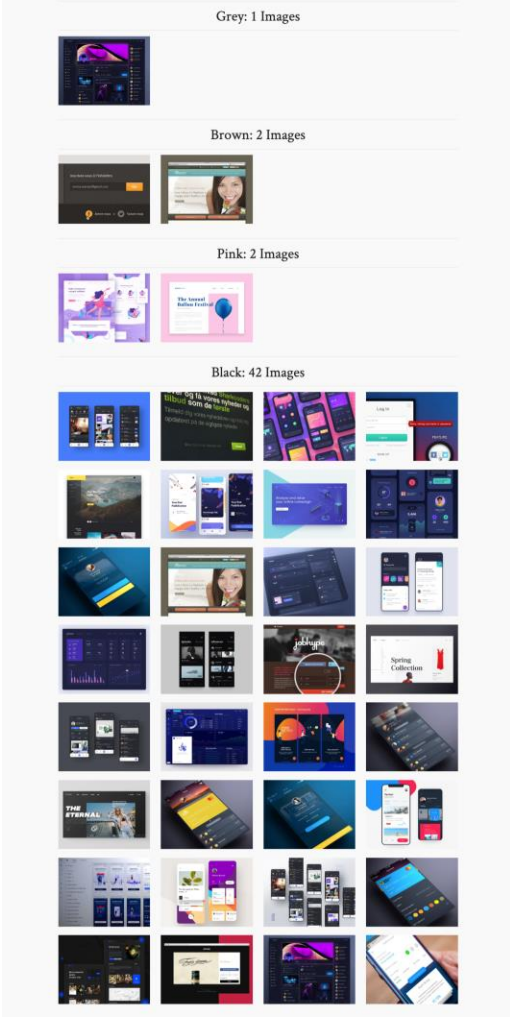
- Exploring Variations
- Quicker Inspiration

Query	Number of Dribbble	Number of our gallery
ios trip list	14	**187**
iphone blue checkout	9	**39**
black finance chart	10	**50**
website music list	3	**16**

Informal Feedback from Designers

- Design Demographic
  - Social => Blue, White

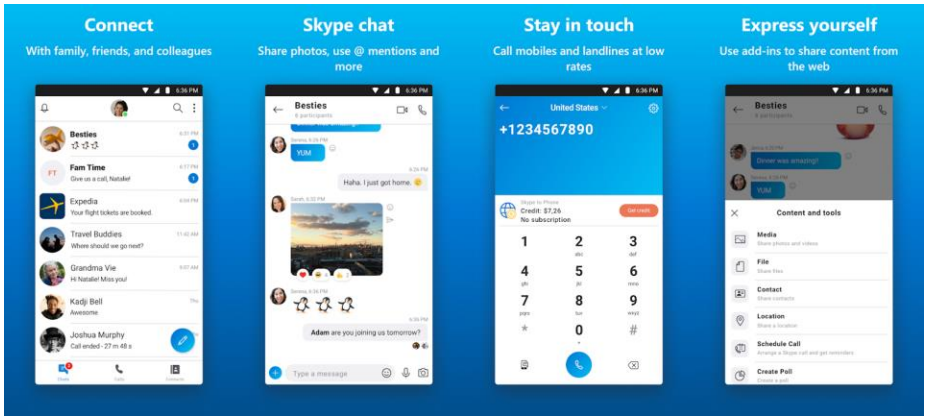
	Number of designs	Proportion
Grey	1	0.51%
Brown	2	1.01%
Pink	2	1.01%
Black	42	21.21%
White	48	24.24%
Green	10	5.05%
Blue	80	40.40%
Red	5	2.53%
Yellow	8	4.04%

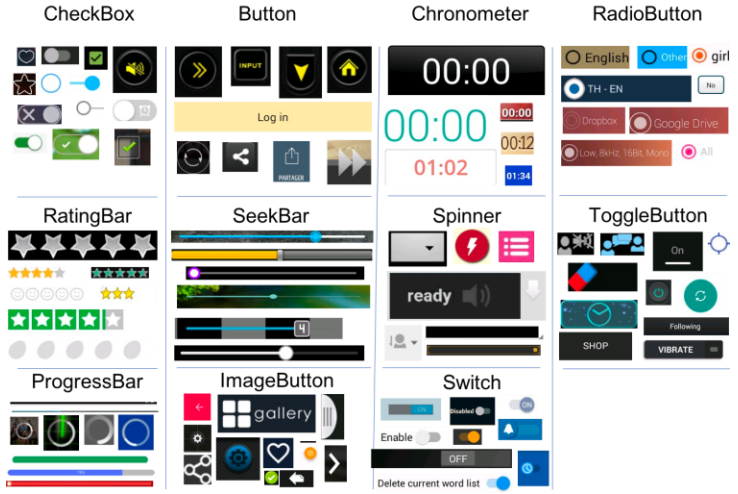


Half-Way

Component-Oriented Search

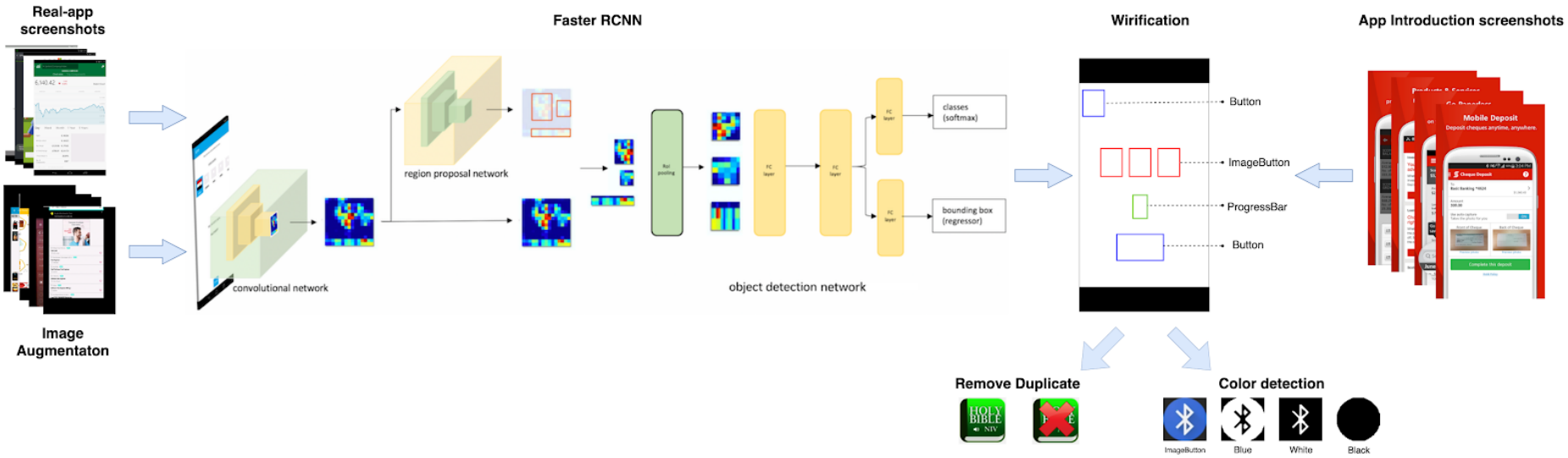
- Problem Definition
  - Exploit app introduction screenshots
  - 11 types of GUI components in Android





System Flow

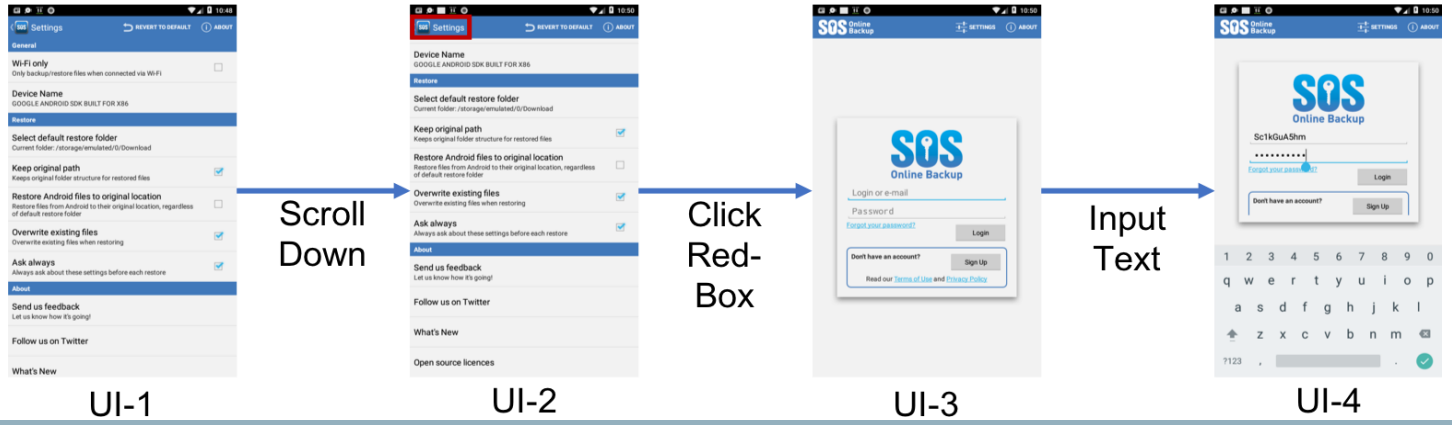
1. Screenshots collection (**real-app**, **app introduction**)
- Components detection (**Faster RCNN** [7])
1. Screenshots **Wirification**
2. Post-processing (**duplicates removal**, **color detection**)



Screenshot Collection

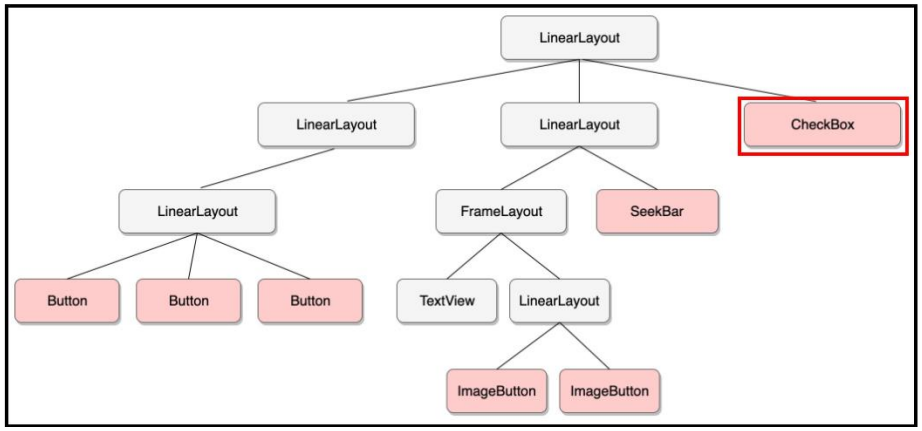
Dataset	Method	Number of screenshots
Real-app	Automated GUI Exploration [8]	68,702
App Introduction	Crawling	469,177

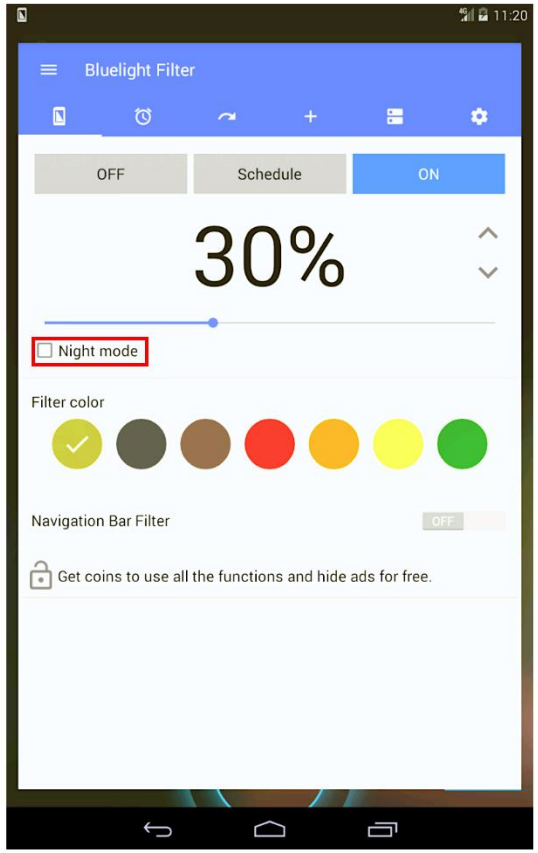
Automated GUI Exploration



Screenshot Collection

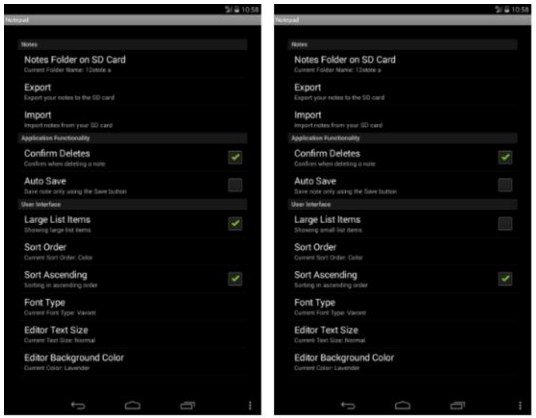
```
...
<node text="" class="android.widget.LinearLayout" bounds="[0,183][800,563]">
    <node text="" class="android.widget.LinearLayout" bounds="[21,204][779,278]">
        <node text="" class="android.widget.LinearLayout" bounds="[21,204][779,278]">
            <node text="OFF" class="android.widget.Button" bounds="[26,209][269,273]" />
            <node text="Schedule" class="android.widget.Button" bounds="[279,209][522,273]" />
            <node text="ON" class="android.widget.Button" bounds="[532,209][774,273]" />
        </node>
    </node>
    <node text="" class="android.widget.LinearLayout" bounds="[21,278][779,499]">
        <node text="" class="android.widget.FrameLayout" bounds="[21,278][779,456]">
            <node text="30%" class="android.widget.TextView" bounds="[276,278][523,456]" />
            <node text="" class="android.widget.LinearLayout" bounds="[715,303][779,431]">
                <node text="" class="android.widget.ImageButton" bounds="[715,303][779,367]" />
                <node text="" class="android.widget.ImageButton" bounds="[715,367][779,431]" />
            </node>
        </node>
        <node text="" class="android.widget.SeekBar" bounds="[21,456][779,499]" />
    </node>
    <node text="Night mode" class="android.widget.CheckBox" bounds="[21,499][191,542]" />
</node>
...
```



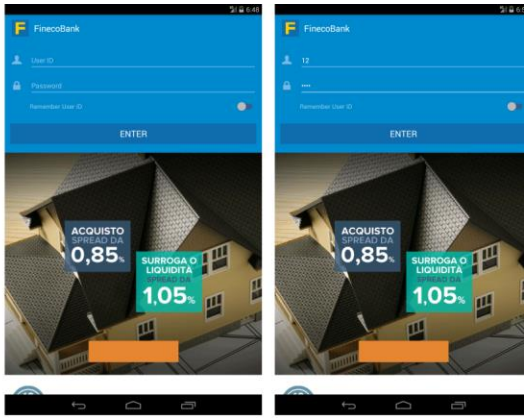


Screenshot Pre-Processing

- Duplicates:
  - Design consistency
  - App policy and privacy
  - Similar structure

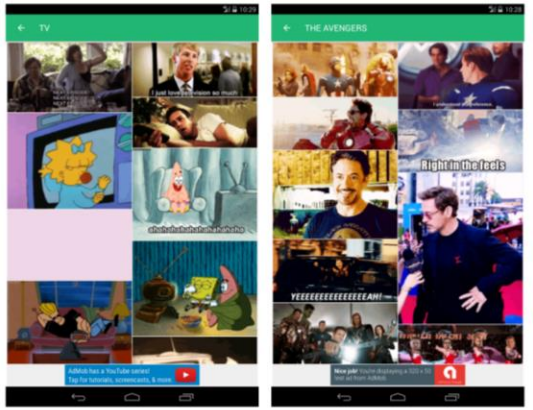


Design consistency



App policy and privacy

- Solution
  - Breadth-First Traversal [9] of DOM tree
    - Eg. *CheckBox[21,499][191,542]SeekBar[21,456]….*
  - Levenshtein distance [10] to calculate string similarity

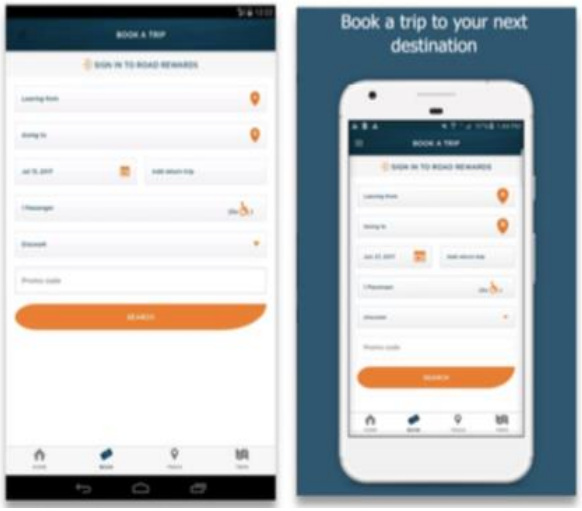


Similar structure

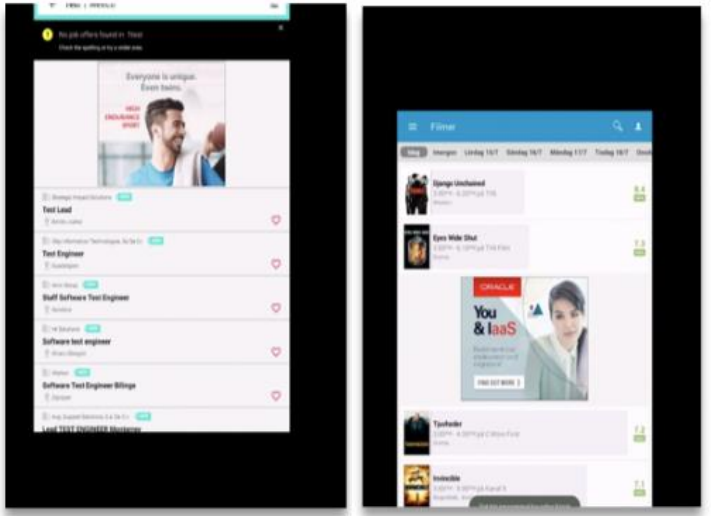
Image Augmentation

- **Problem**: App introduction screenshots contains other information
- **Solution**: randomly **resize** and **move around** over center

Problem



Solution



Evaluation

- Evaluation Metrics
  - Recall, Precision, mAP based on IoU

With image augmentation

IoU	Recall	Precision	mAP
0.6	0.65	0.73	0.69
0.7	0.60	0.79	0.66
0.8	0.53	0.84	0.62
0.9	0.36	0.90	0.44

Without image augmentation

IoU	Recall	Precision	mAP
0.6	0.51	0.68	0.56
0.7	0.44	0.72	0.48
0.8	0.37	0.80	0.43
0.9	0.26	0.87	0.36

Overall performance, **Recall**: *0.62*, **Precision**: *0.76*, **mAP**: *0.51*

Evaluation

- Model Behaviour

(a) Components with texts in different languages (accurately detected)

(b) Components sharing the same background (accurately detected)

(c) Many components (accurately detected)

Without Augmentation

With Augmentation

Without Augmentation

With Augmentation

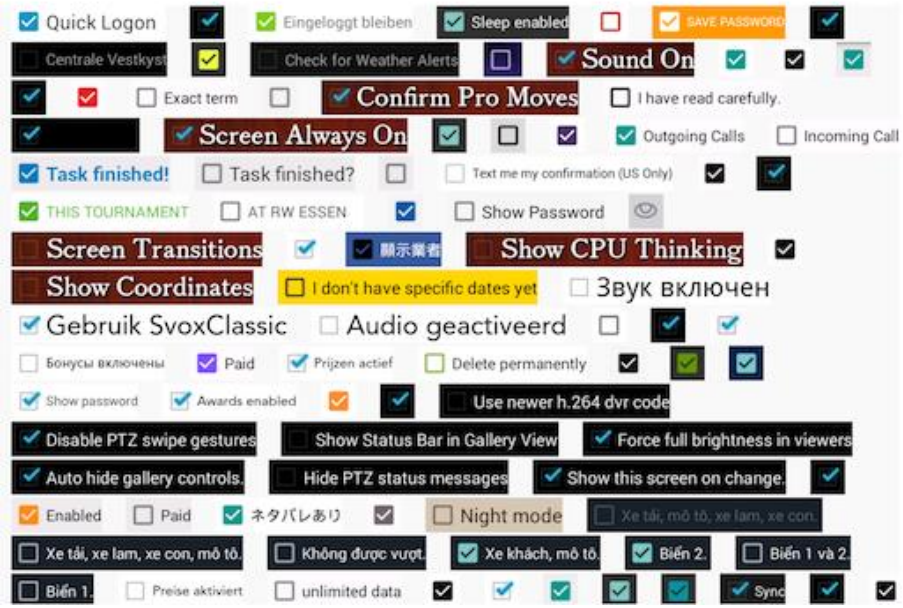
Post-processing GUI Components

Removing Duplicate UI Components	**Structural Similarity Index (SSIM)** [11]
Determining Primary Color	**HSV color space** [12]
Font Indentification	**Myfonts tool** [13]

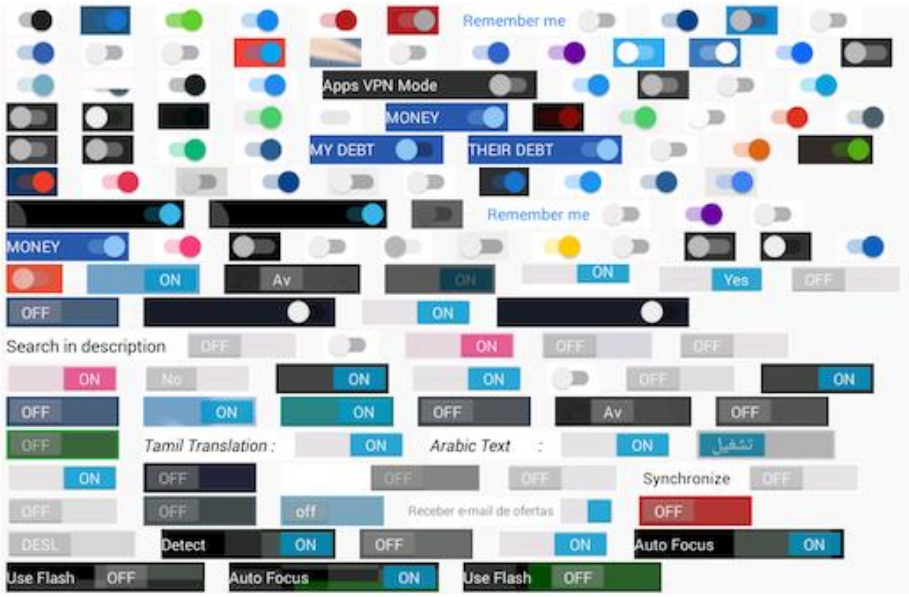
Post-processing GUI Components

- Removing Duplicate UI Components

CheckBox

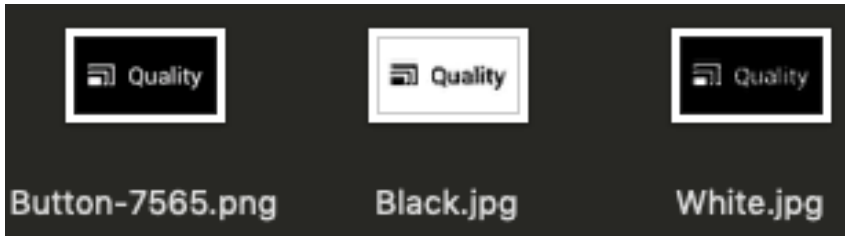


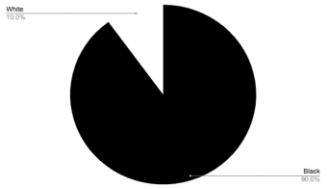
Switch

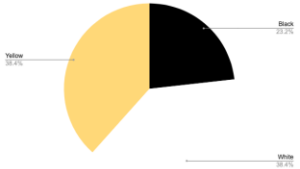


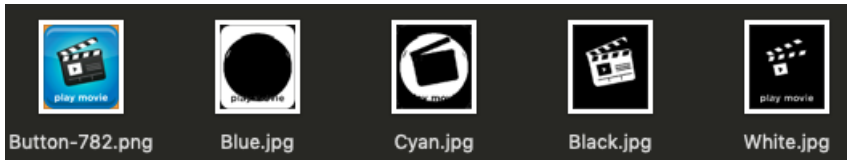
Post-processing GUI Components

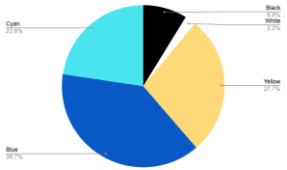
- Determining Primary Color

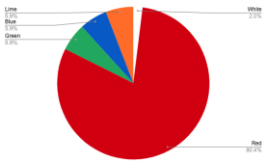








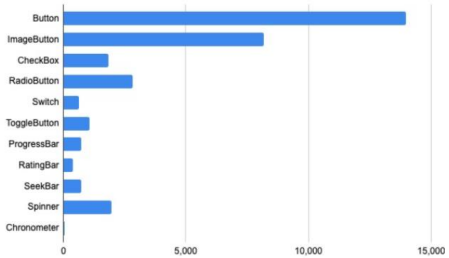


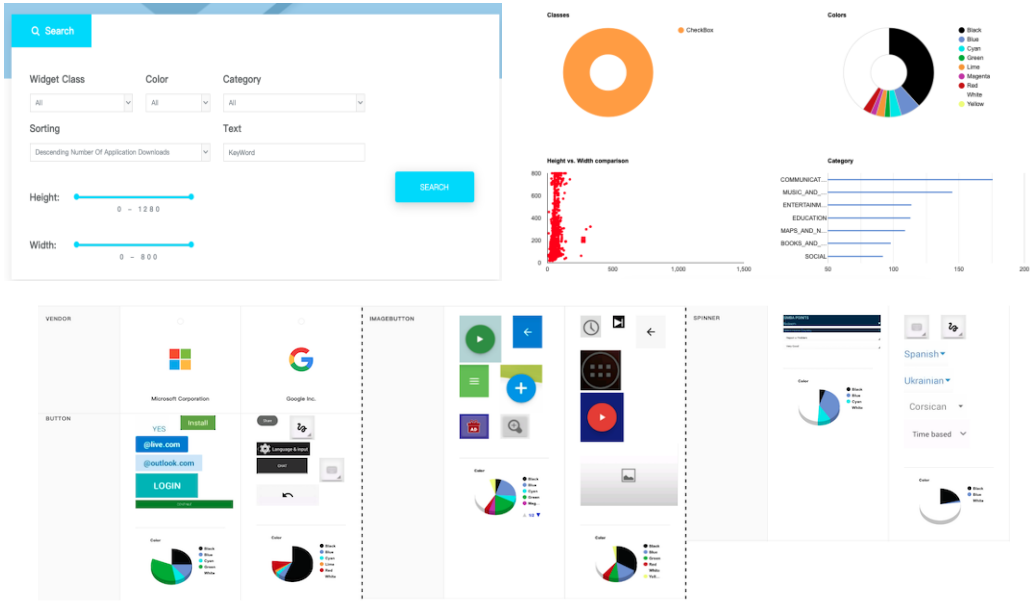


Design Gallery

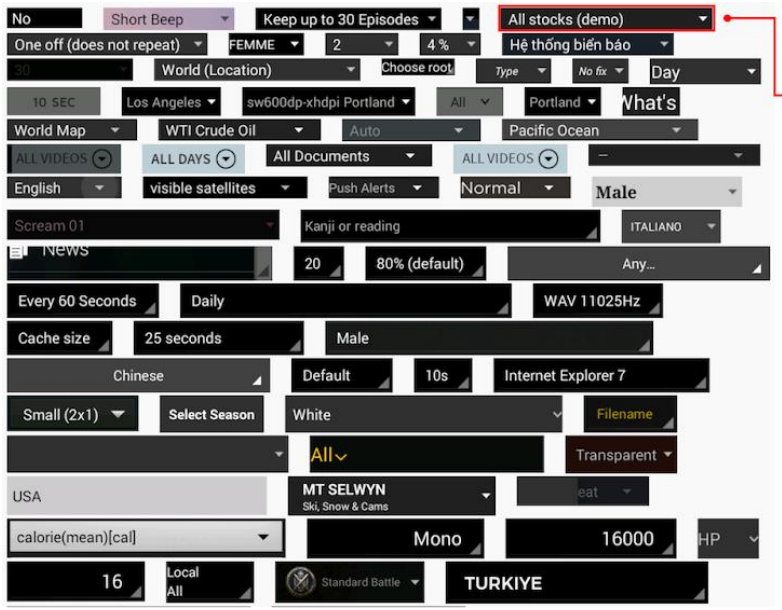
- Database
- Multi-faceted Search
- Design Demographics
- Design Comparison

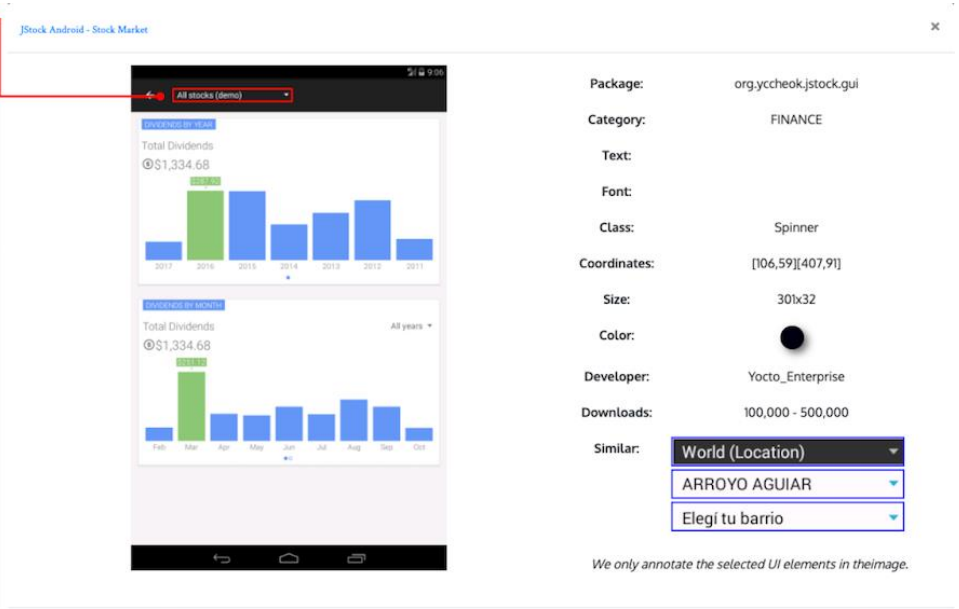
	Size	Proportion
Button	13,970	43.28%
ImageButton	8,182	25.35%
CheckBox	1,838	5.69%
RadioButton	2,803	8.68%
Switch	614	1.90%
ToggleButton	1,045	3.24%
ProgressBar	707	2.19%
RatingBar	392	1.21%
SeekBar	728	2.26%
Spinner	1,966	6.09%
Chronometer	33	0.10%





Design Gallery



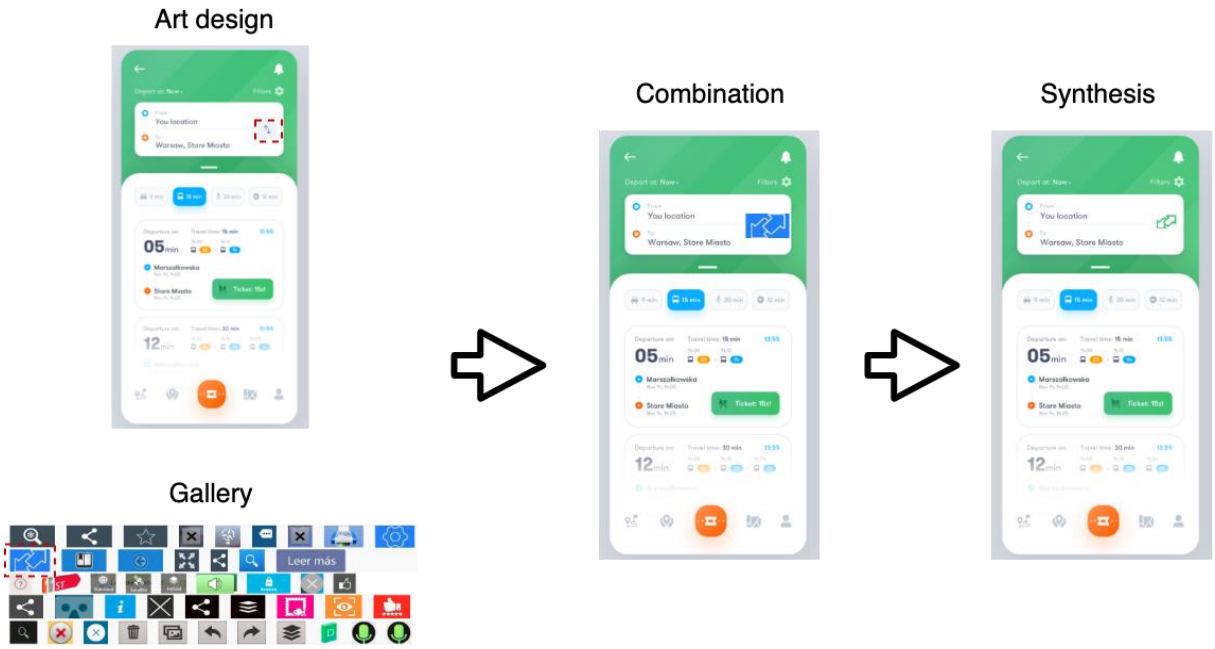


Conclusion

We provide a GUI Design Gallery through invisible crowdsourcing: supporting design search and knowledge discovery beyond content sharing

Future work

- Support image-based search
- Extend gallery database, including video analysis
- Design Synthesis



Corporation

Reference

[1] Najork, M. and Wiener, J. L., 2001. Breadth-first crawling yields high-quality pages. In *Proceedings of the 10th international conference on World Wide Web*, 114–118. ACM.

[2] Dribbble, 2010. Dribbble - discover the world's top designers & creatives. https://dribbble.com/

[3] Agrawal, R.; Srikant, R.; et al., 1994. Fast algorithms for mining association rules. In*Proc.20thint.conf.verylargedatabases,VLDB*,vol.1215,487–499.

[4] Inokuchi, A.; Washio, T.; and Motoda, H., 2000. An apriori-based algorithm for mining frequent substructures from graph data. In *European conference on principles ofdataminingandknowledgediscovery*,13–23.Springer.

[5] Chen, C.; Xing, Z.; and Wang, X., 2017b. Unsupervised software-specific morpho- logical forms inference from informal discussions. In *Proceedings of the 39th Inter- national Conference on Software Engineering*, 450–461. IEEE Press.

[6] Loper, E. and Bird, S., 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*,(2002).

[7] Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processingsystems*,91–99.

[8] Su, T.; Meng, G.; Chen, Y.; Wu, K.; Yang, W.; Yao, Y.; Pu, G.; Liu, Y.; and Su, Z., 2017. Guided, stochastic model-based gui testing of android apps. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 245–256. ACM.

[9] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C., 2009. *Introduction to algorithms*.MITpress. (citedonpage42)

Reference

[10] Levenshtein, V. I., 1966. Binary codes capable of correcting deletions, insertions, andreversals. In *Sovietphysicsdoklady*, vol. 10, 707–710.

[11] Hore, A. and Ziou, D., 2010. Image quality metrics: Psnr vs. ssim. In *2010 20th InternationalConferenceonPatternRecognition*, 2366–2369. IEEE.

[12] Muhammad, B. and Abu-Bakar, S. A. R., 2015. A hybrid skin color detection using hsv and ycgcr color space for face detection. In *2015 IEEE International Conference onSignalandImageProcessingApplications(ICSIPA)*, 95–98. IEEE.

[13] 2019. https://www.myfonts.com/WhatTheFont.

Reliability-Aware Service Function Chain Provisioning In Mobile Edge-Cloud Networks

Shouxu Lin

u5898871

Supervisor: Prof. Weifa Liang

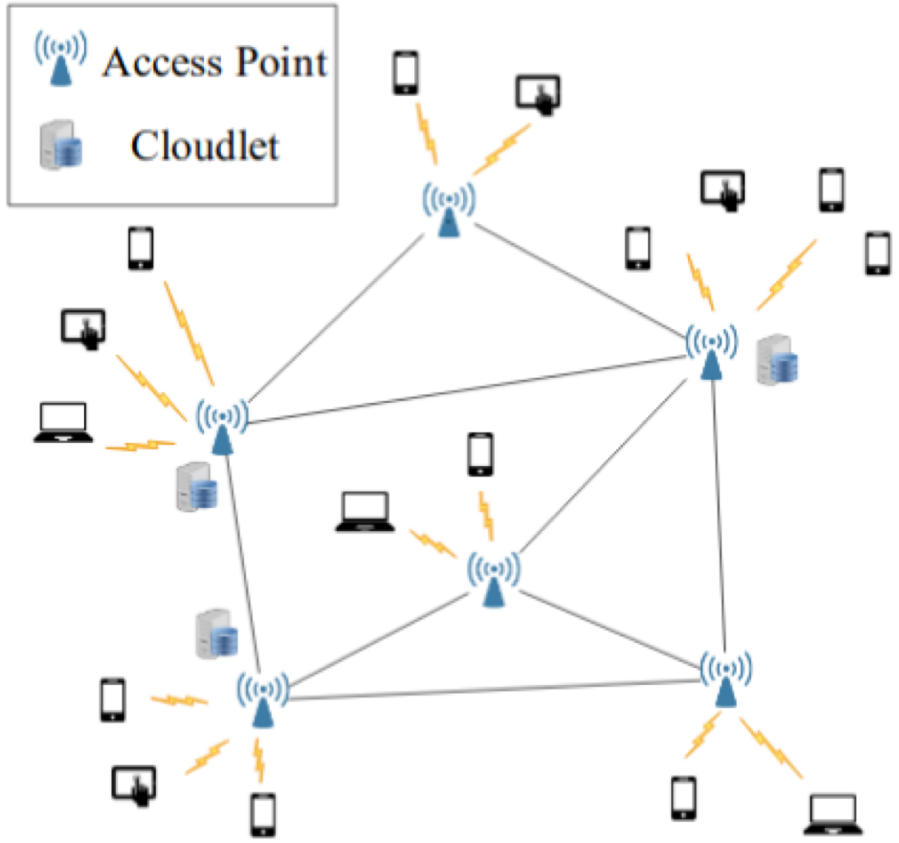
OUTLINE

- Background
  - Mobile Edge Computing (MEC)
  - Virtualized Network Function (VNF)
  - Service Function Chain (SFC)
  - Reliability Issue
  - Backup Strategy
- Related Work
- Problem Formulation & NP-hardness
- Algorithms
  - Integer Linear Programming (ILP)
  - Randomized Algorithm
  - Heuristic Algorithm
- Performance Evaluation

Background – Mobile Edge Computing (MEC)

- Definition: A network architecture

- Motivation:
  - Mobile devices have limited computing capacity and battery capacity
  - Leverage the capability of mobile devices through offloading their tasks to the MEC



Background – Virtualized Network Function (VNF)

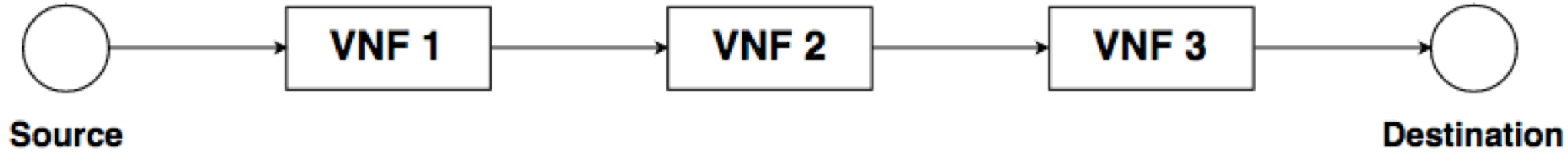
- Definition: Implements various network functions by pieces of software running in virtual machines of commodity servers

- Motivation:
  - Replace dedicated hardware appliances with VNF instances
  - Make the network easy to update
  - Reduce cost

Background – Service Function Chain (SFC)

Definition:

Service Function Chain (SFC): consist of several different types of VNFs.



Background – Reliability Issue

- Definition: the ability for the network to provide and maintain an acceptable level of service in the face of failures and challenges to normal operation
- Traditional carried-grade system: 99.999% reliability (roughly 25.9 seconds downtime per month)
- Difficult for VNF to achieve this level of reliability

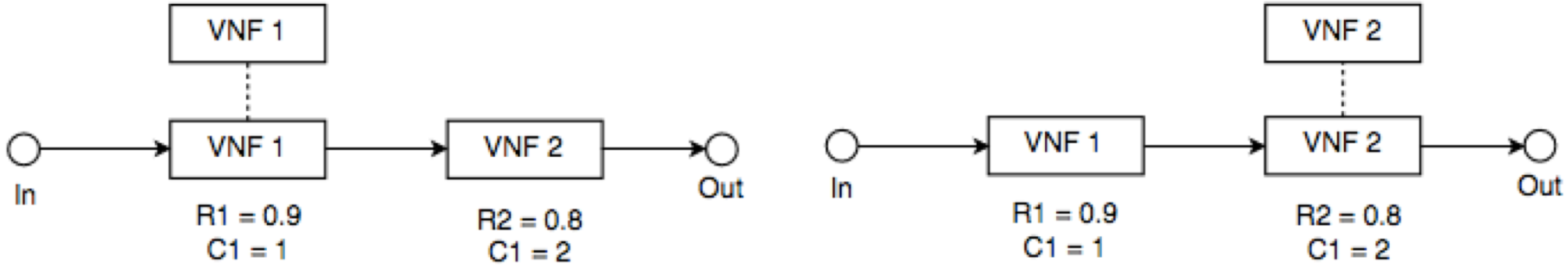
Background – Backup Strategy

- Motivation: Additional VNF instances are placed for the backup purpose.
- Reliability
  - VNF: the probability that at least one of its instances is available
  - SFC: the probability that at least one instance of each of its VNF is available

Challenges

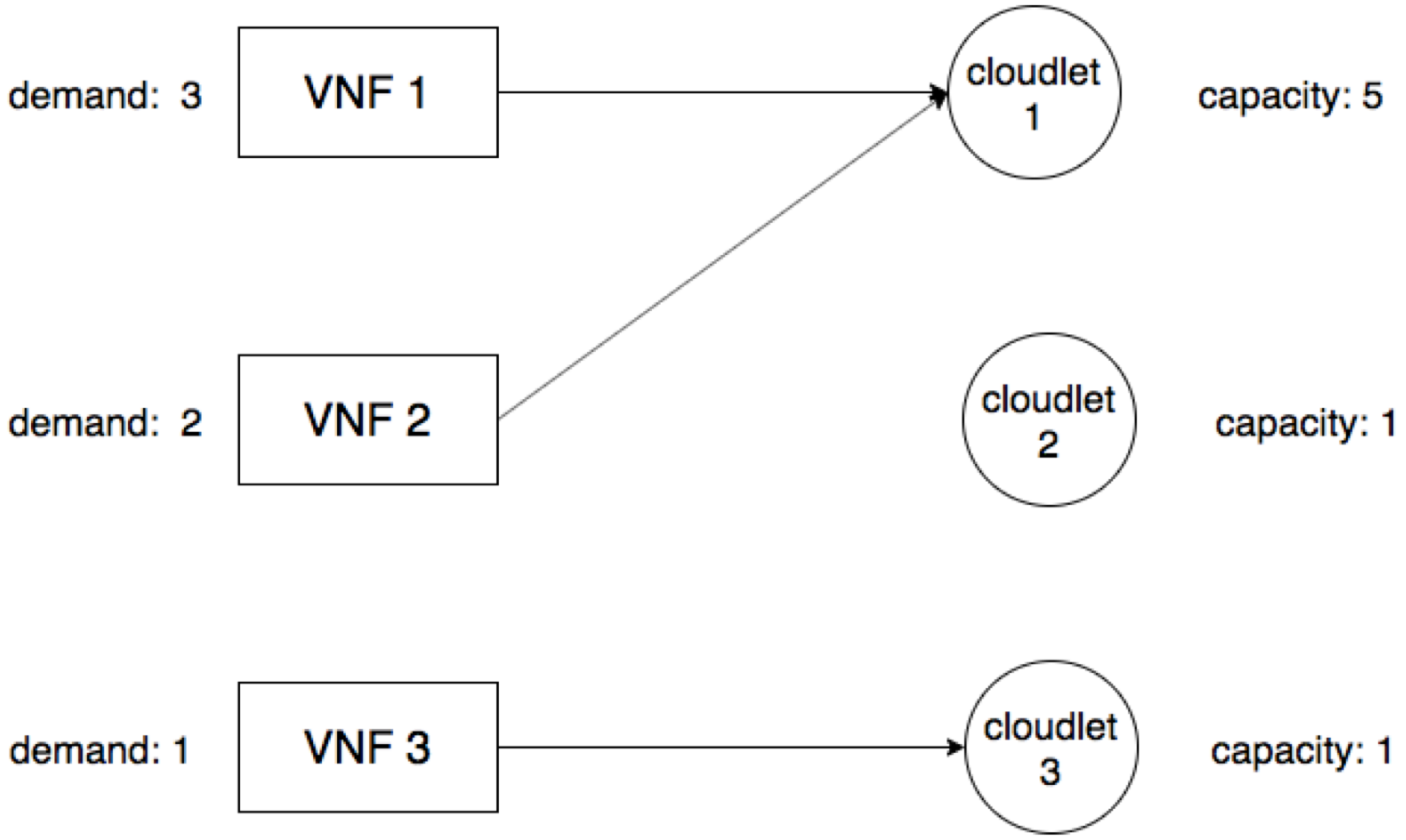
- Determining the number of instances required for each VNF
- Place the VNF instances to cloudlets

Challenges



	Current Reliability θ	New Reliability θ'	Increase in reliability θ'-θ	Computing resource c
VNF 1	0.9 * 0.8 = 0.72	0.99 * 0.8 = 0.792	0792-0.72=0.072	1
VNF 2	0.9 * 0.8 = 0.72	0.9 * 0.96 = 0.864	0.864-0.72 = 0.144	2

Challenges


demand: 3
VNF 1
cloudlet 1
capacity: 5
demand: 2
VNF 2
cloudlet 2
capacity: 1
demand: 1
VNF 3
cloudlet 3
capacity: 1

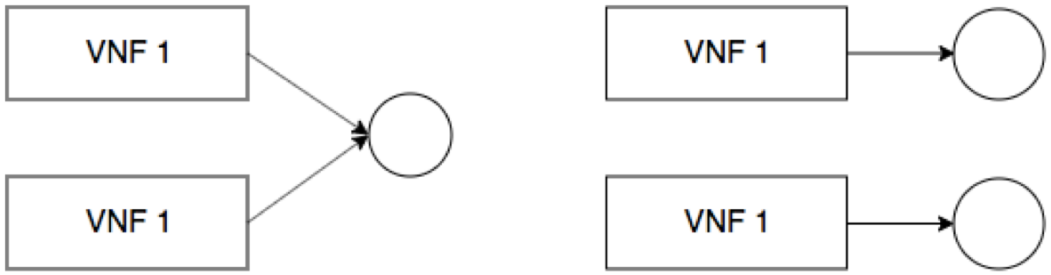
Related Work

- Assumption:
  - There is only one backup instance for each VNF.
  - Different SFC requests have the same reliability requirements.
  - The VNF instances of all network functions must be placed into one cloudlet (server).
  - There are unlimited computing resource for VNF placement.

Problem Formulation

- Problem:
  - Admit as many as requests
  - Satisfy the specified reliability requirement of each admitted request

- Assume that for a specific request, all the instances of a specific VNF must be placed to the same cloudlet.



NP-hardness

- NP-hardness of the defined problem:
  - Reduced from the well-known NP problem, Generalized Assignment Problem (GAP).

Integer Linear Programming (ILP)

$$\textit{Maximize} \quad \sum_{r_k \in R} y_k,$$

subject to:

$$\sum_{r_k \in R} \sum_{f_i \in SFC_k} \sum_{l \in W} x_{k,i,l,j} \cdot c_{k,i,l} \leq cap_j, \qquad \forall cl_j \in \mathcal{C}$$

$$\sum_{f_i \in SFC_k} \sum_{l \in W} \sum_{cl_j \in C} x_{k,i,l,j} \cdot p_{k,i,l} \leq p_k, \qquad \forall r_k \in R$$

$$\sum_{l \in W} \sum_{cl_j \in C} x_{k,i,l,j} = y_k, \qquad \forall r_k \in R, f_i \in SFC_k$$

$$y_k \in \{0,1\}, \qquad \forall r_k \in \mathcal{R}$$

$$x_{k,i,l,j} \in \{0,1\} \qquad \forall r_k \in R, f_i \in SFC_k, l \in W, cl_j \in C$$

Integer Linear Programming (ILP)

- Constraint 1 ensures that the total computing resource demanded by all the VNF instances implemented in any cloudlet does not exceed the cloudlet's computing capacity.

- Constraint 2 ensures that the reliability requirement of each request needs to be met if the request is admitted.

- Constraint 3 says that if any request is admitted, then at least one instance of each VNF should be implemented, and all the instances of the same VNF should be implemented at the same cloudlet.

- Constraint 4 & 5 limit the range of each boolean variables to either 0 or 1.

Integer Linear Programming (ILP)

			Placed at Cloudlet 1	Placed at Cloudlet 2
Request 1 $y_1 = 1$	VNF 1	Has 1 instance	$x_{1,1,1,1} = 0$	$x_{1,1,1,2} = 1$
		Has 2 instances	$x_{1,1,2,1} = 0$	$x_{1,1,2,2} = 0$
	VNF 2	Has 1 instance	$x_{1,2,1,1} = 0$	$x_{1,2,1,2} = 0$
		Has 2 instances	$x_{1,2,2,1} = 1$	$x_{1,2,2,2} = 0$
Request 2 $y_2 = 0$	VNF 1	Has 1 instance	$x_{2,1,1,1} = 0$	$x_{2,1,1,2} = 0$
		Has 2 instances	$x_{2,1,2,1} = 0$	$x_{2,1,2,2} = 0$
	VNF 2	Has 1 instance	$x_{2,2,1,1} = 0$	$x_{2,2,1,2} = 0$
		Has 2 instances	$x_{2,2,2,1} = 0$	$x_{2,2,2,2} = 0$

Randomized Algorithm

- Overview:
  - Perform Linear Programming (LP) relaxation of the ILP
  - Randomized Rounding
    - E.g. $x = 0.8 \rightarrow$ set $x = 1$ with probability 0.8 and 0 otherwise

Randomized Algorithm

			Placed at Cloudlet 1	Placed at Cloudlet 2
Request 1 $y_1 = 0.8$	VNF 1	Has 1 instance	$x_{1,1,1,1} = 0.2$	$x_{1,1,1,2} = 0.2$
		Has 2 instances	$x_{1,1,2,1} = 0.3$	$x_{1,1,2,2} = 0.1$
	VNF 2	Has 1 instance	$x_{1,2,1,1} = 0.4$	$x_{1,2,1,2} = 0.1$
		Has 2 instances	$x_{1,2,2,1} = 0.1$	$x_{1,2,2,2} = 0.1$
Request 2 $y_2 = 0.2$	VNF 1	Has 1 instance	$x_{2,1,1,1} = 0.1$	$x_{2,1,1,2} = 0.04$
		Has 2 instances	$x_{2,1,2,1} = 0$	$x_{2,1,2,2} = 0.06$
	VNF 2	Has 1 instance	$x_{2,2,1,1} = 0.05$	$x_{2,2,1,2} = 0.06$
		Has 2 instances	$x_{2,2,2,1} = 0.03$	$x_{2,2,2,2} = 0.06$

Randomized Algorithm – Approximation Ratio

- The approximation ratio of the randomized algorithm: $\alpha = \sqrt{\frac{2 \log |R|}{\tilde{\mu}}}$

Randomized Algorithm – Constraint Violation

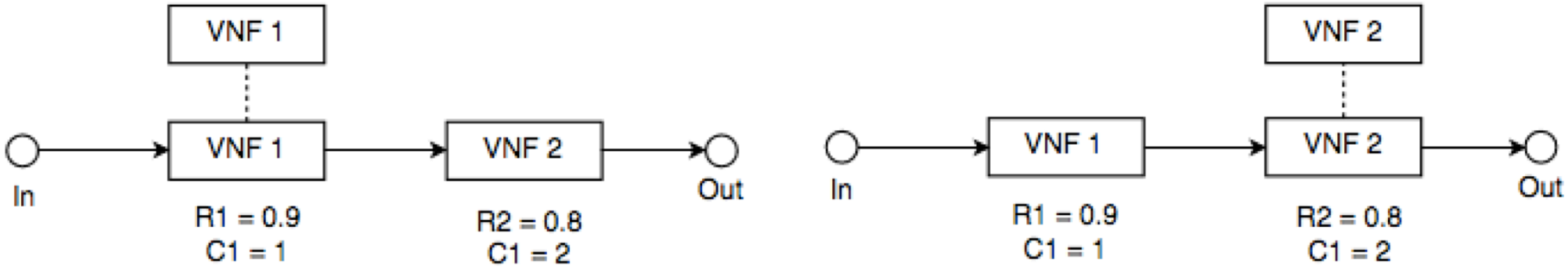
$$\sum_{r_k \in R} \sum_{f_i \in SFC_k} \sum_{l \in W} x_{k,i,l,j} \cdot c_{k,i,l} \leq cap_j, \qquad \forall cl_j \in \mathcal{C}$$

- Computing capacity constraint violation < $\sigma = \frac{\log|C| + \sqrt{\log^2|C| + 4\lambda \log|C|}}{\lambda}$

$$\sum_{f_i \in SFC_k} \sum_{l \in W} \sum_{cl_j \in C} x_{k,i,l,j} \cdot p_{k,i,l} \leq p_k, \qquad \forall r_k \in R$$

- Reliability constraint violation < $\varepsilon = \frac{\log|R| + \sqrt{\log^2|R| + 4\lambda \log|R|}}{\lambda}$.

Heuristic Algorithm- Determine the Number of Instances

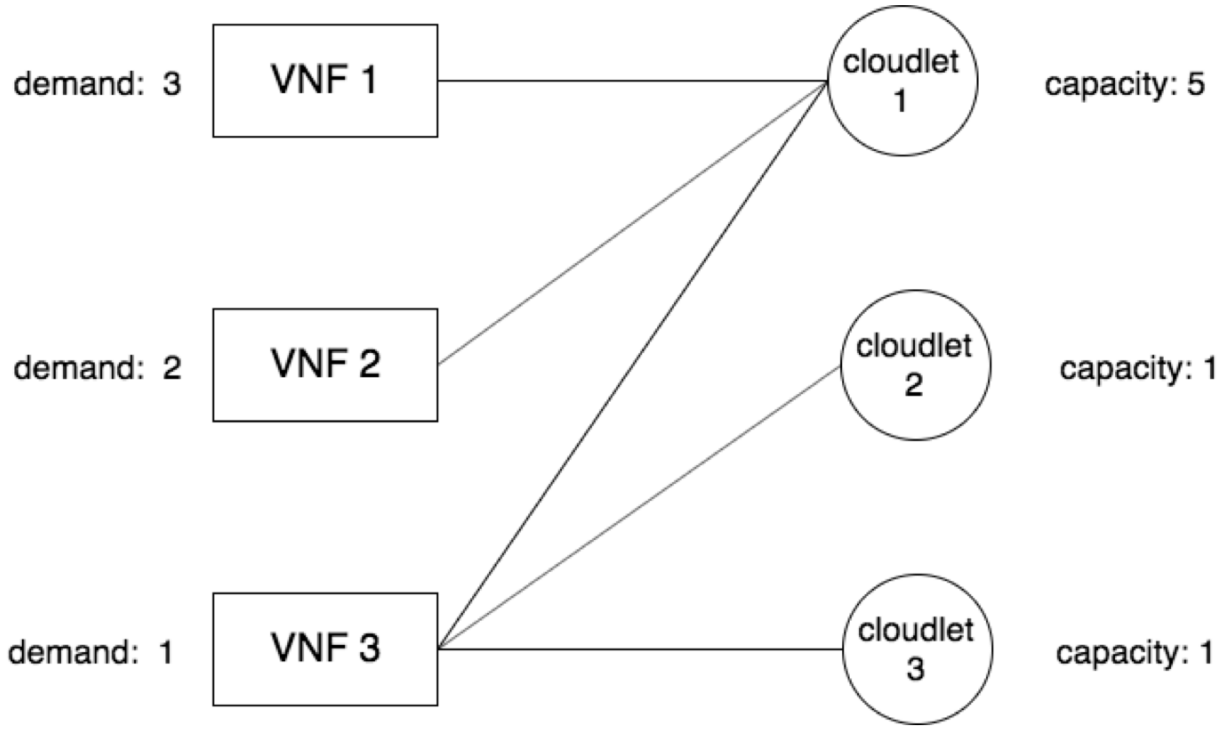


	Current Reliability θ	New Reliability θ’	Increase in reliability θ’-θ	Computing resource c	Ratio $\frac{\theta'-\theta}{c}$
VNF 1	0.9 * 0.8 = 0.72	0.99 * 0.8 = 0.792	0792-0.72=0.072	1	0.072
VNF 2	0.9 * 0.8 = 0.72	0.9 * 0.96 = 0.864	0.864-0.72 = 0.144	2	0.072

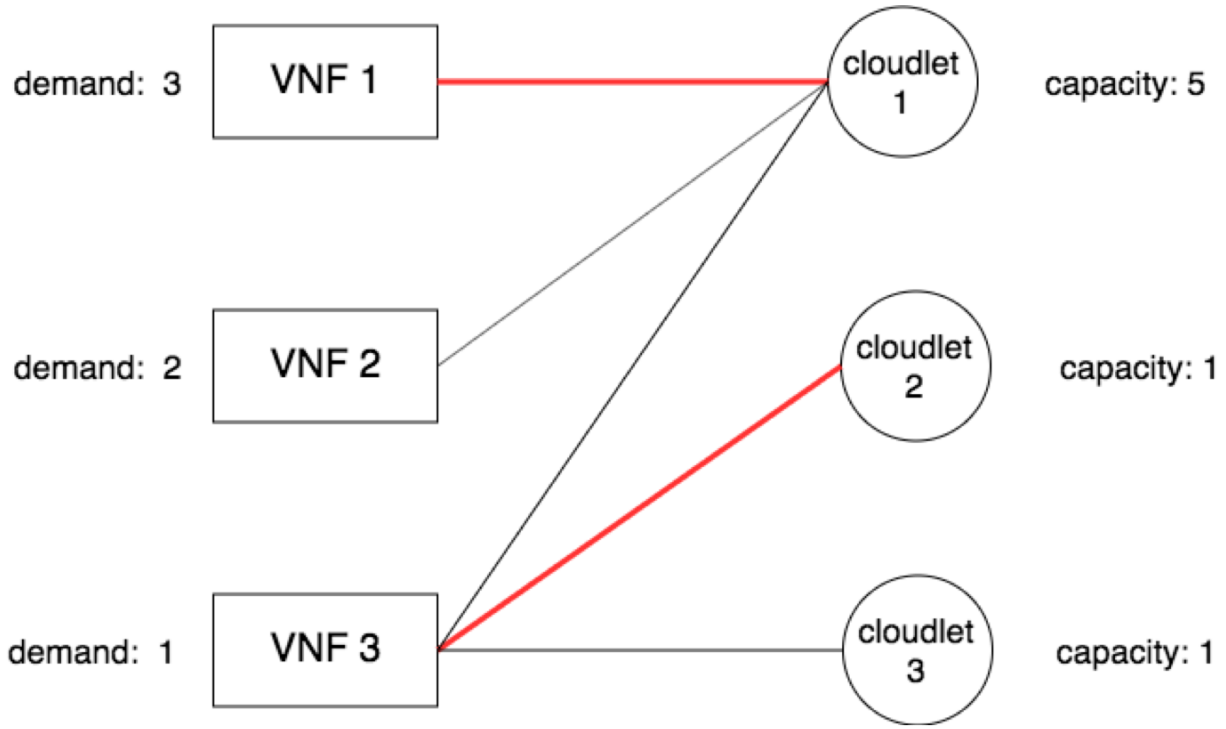
Heuristic Algorithm – VNF Placement

- Place VNFs to cloudlets
  - Construct a Bipartite graph G = (V, E)
    - V: VNFs & cloudlets,
    - E: an edge between a VNF and cloudlet only if the cloudlet has enough residual computing capacity to implement the VNF
    - Find a maximum matching by the Hopcroft-Karp algorithm
  - Repeat Until either
    - All the VNFs haven been placed → admit the request
    - E becomes empty → reject the request

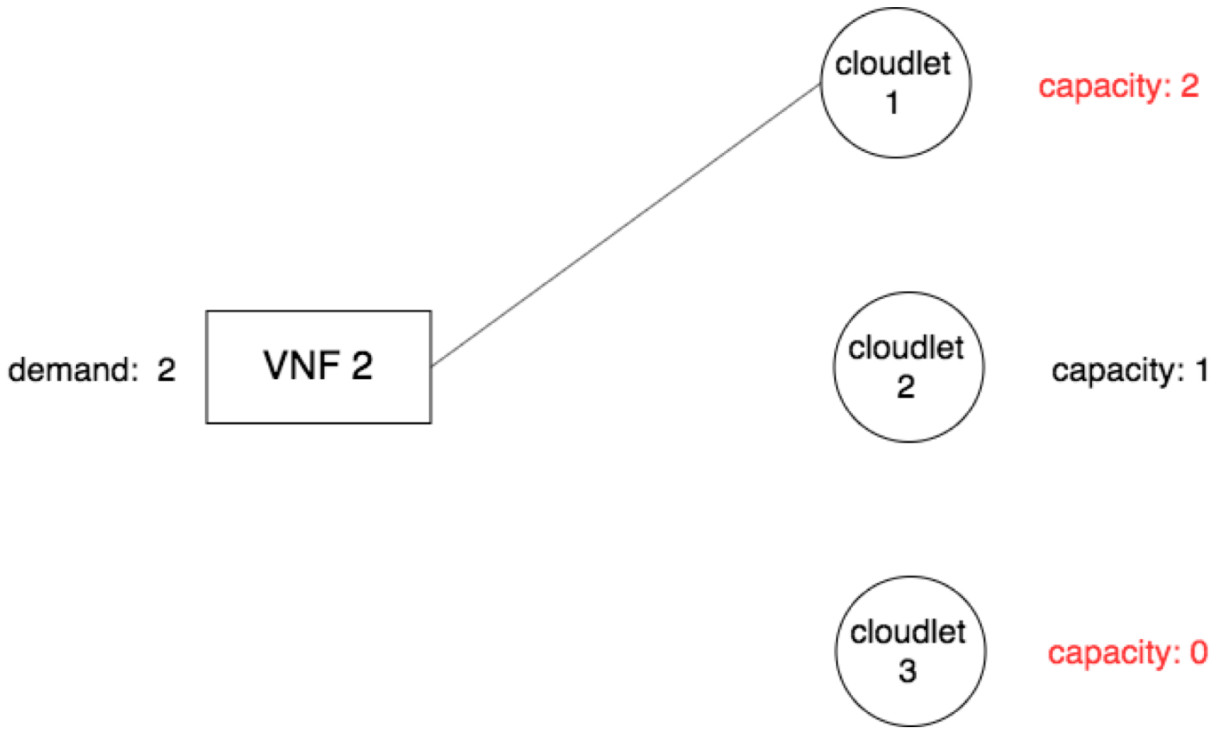
Heuristic Algorithm



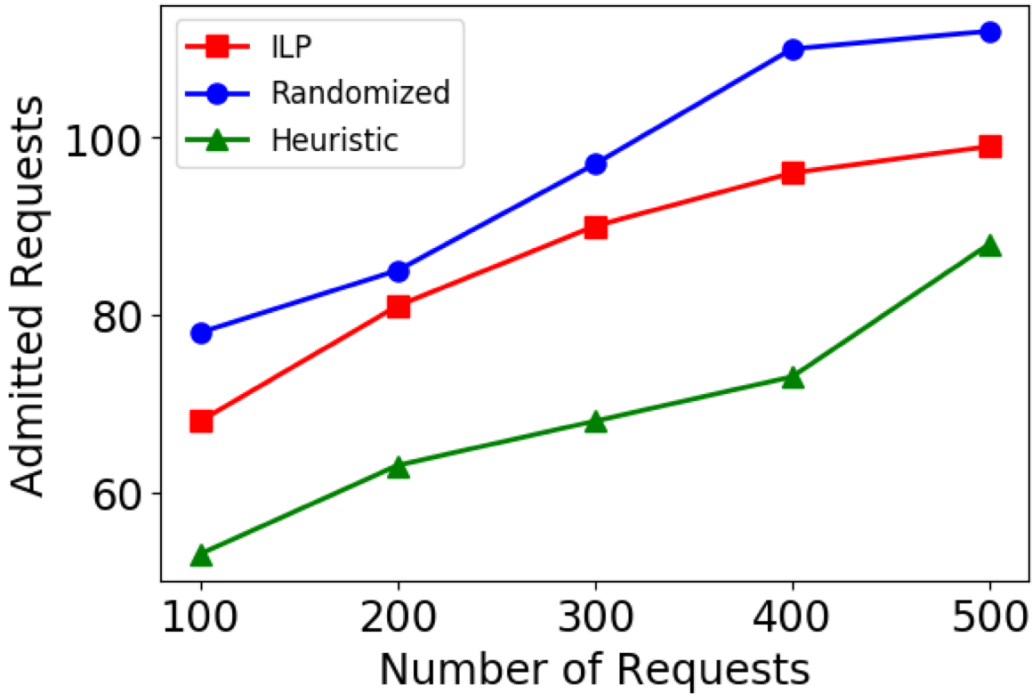
Heuristic Algorithm

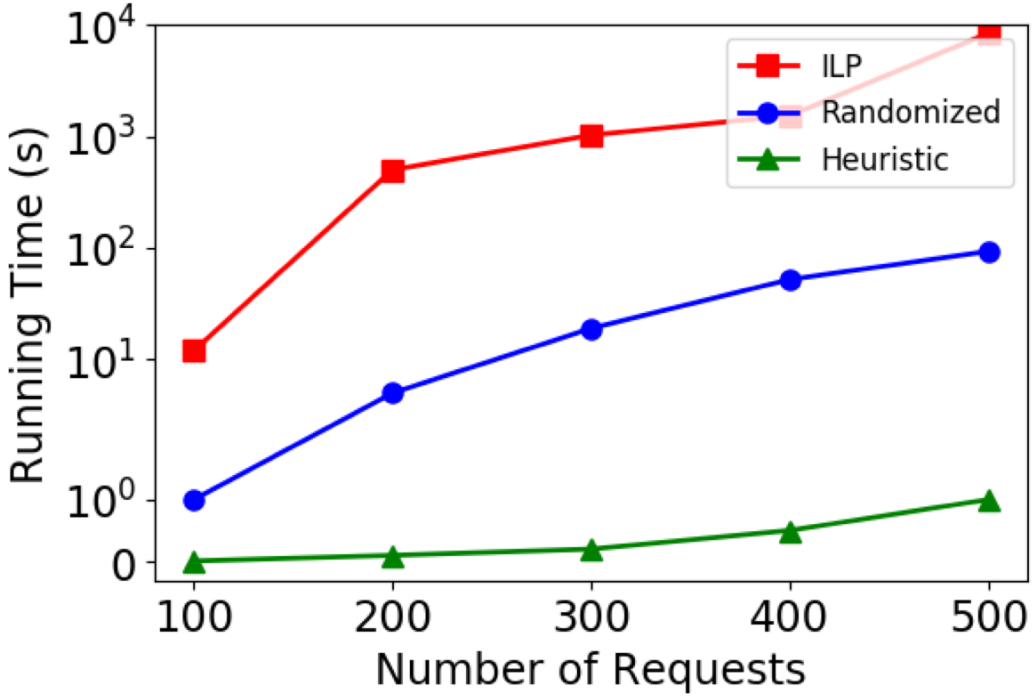


Heuristic Algorithm

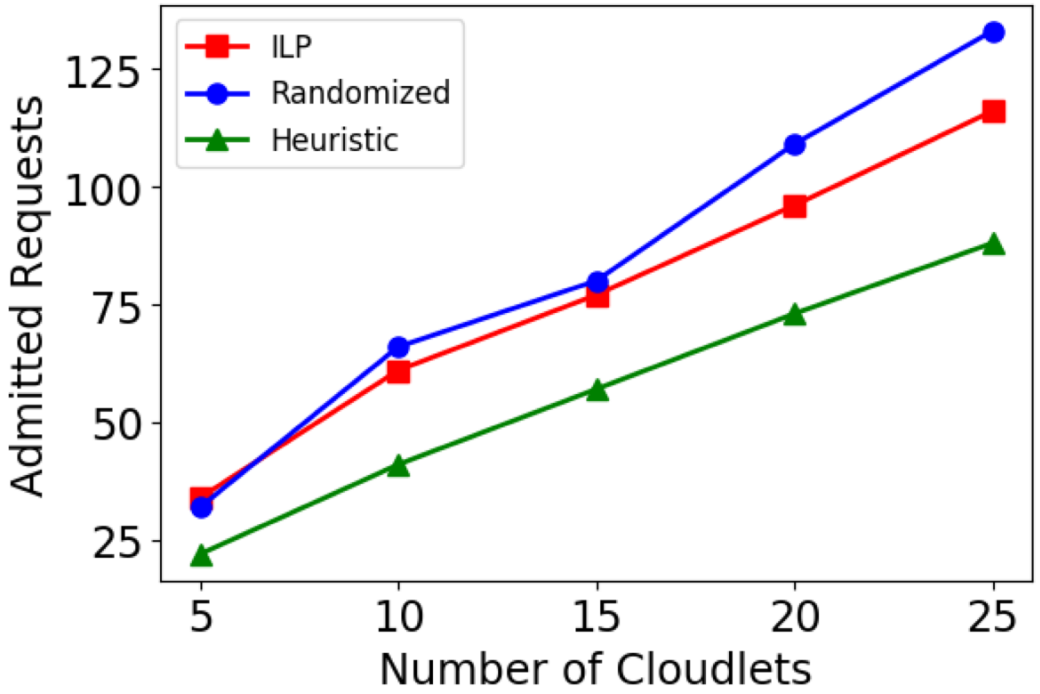


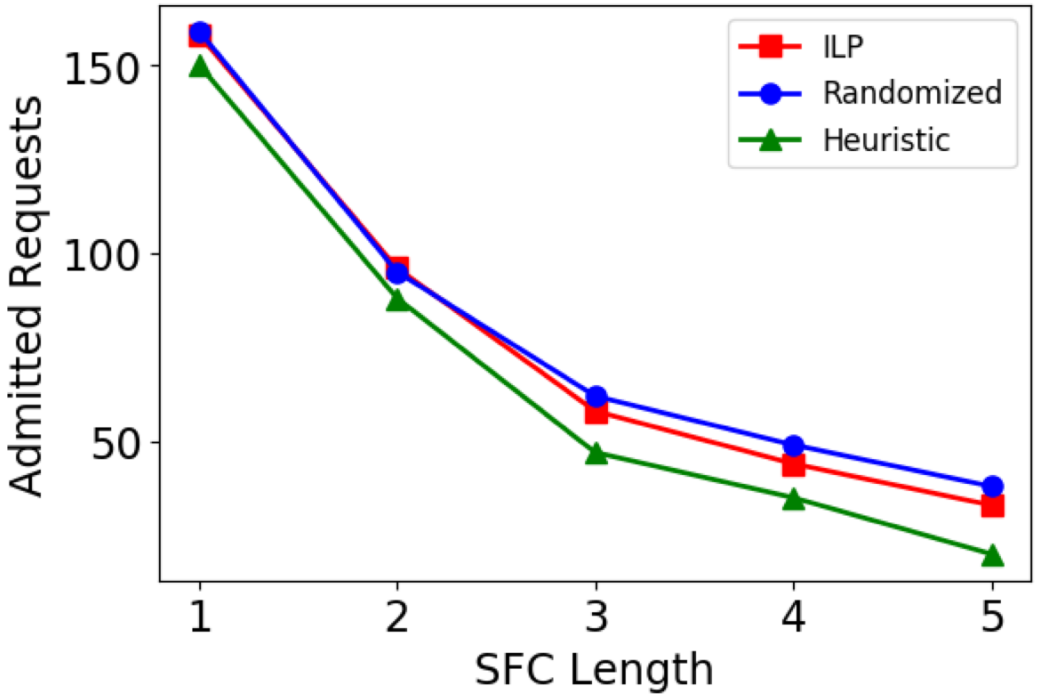
Experiment - Performance Evaluation



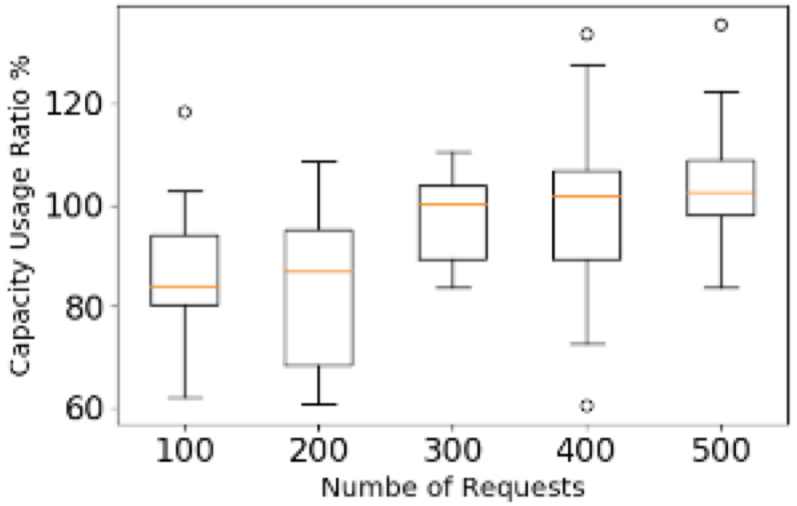


Experiment – Impact of Parameters

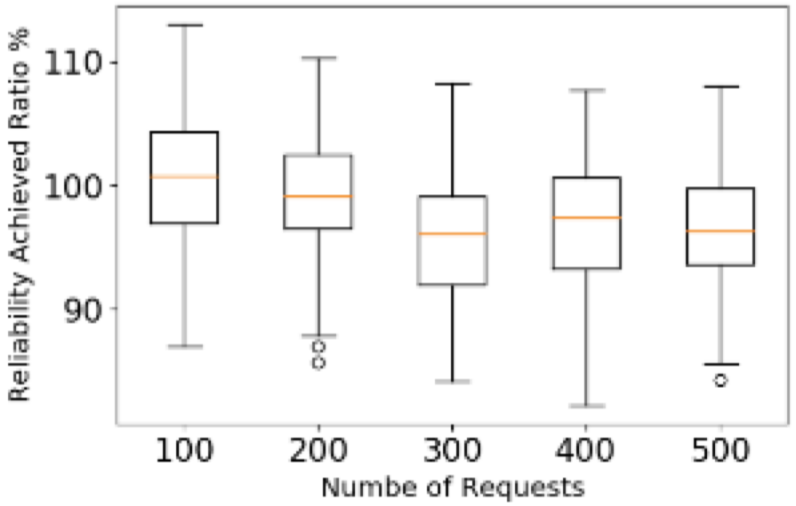




Experiment – Violation Analysis



(a) Capacity constraint violation



(b) Reliability constraint violation

Outcome

- On Submission to IEEE-ICC 2020.

End

Thank you!

Spatial and Temporal Information in Video-based Tasks

Heming Du

Supervisor: Dr. Liang Zheng

Academic Awards

- The winner of the second Micro-Expression Grand Challenge (MEGC) 2019
- The second runner-up in 2019 AI City Challenge Re-ID track
- The fifth place in 2019 AI City Challenge MTMCT track
- The winner of 2019 Mandarin Audio-Visual Speech Recognition Challenge (MAVSR)

Publications

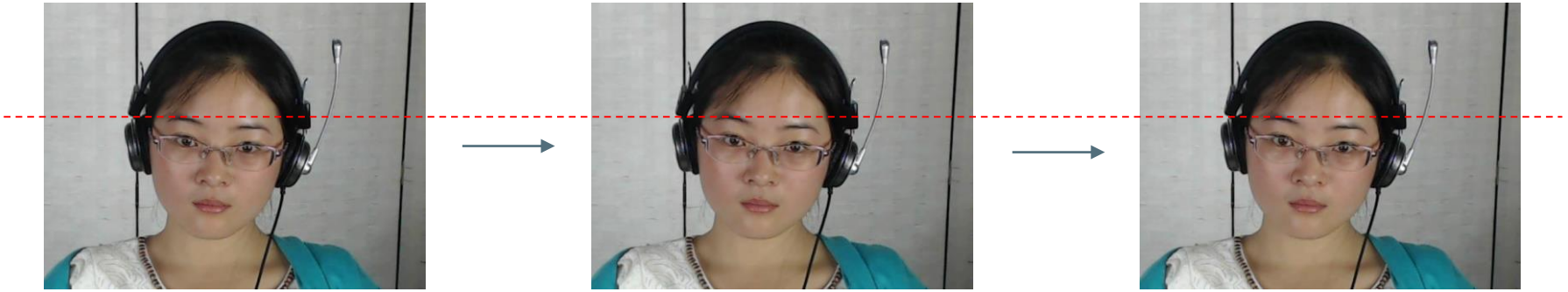
- Liu, Y., Du, H., Zheng, L., & Gedeon, T. (2019, May). A neural micro-expression recognizer. In 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019) (pp. 1-4). IEEE.
- Lv, K., Du, H., Hou, Y., Deng, W., Sheng, H., Jiao, J., & Zheng, L. (2019). Vehicle Re-Identification with Location and Time Stamps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 399-406).
- Hou, Y., Du, H., & Zheng, L. (2019). A Locality Aware City-Scale Multi-Camera Vehicle Tracking System. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 167-174).
- Yao, Y.; Wang, T.; Du, H.; Zheng, L.;andGedeon, T., 2019. Spotting visual keywords from temporal sliding windows. In 2019 International Conference on Multimodal Interaction (pp. 536-539).

Sections

- Micro-expression Recognition
- Vehicle Multi-Target Multi-Camera Tracking
- Vehicle Re-Identification
- Lip Reading Keyword Spotting
- Visual Navigation

Facial Micro-Expression Recognition

Facial Micro-Expression



Onset Frame

Apex Frame

Offset Frame

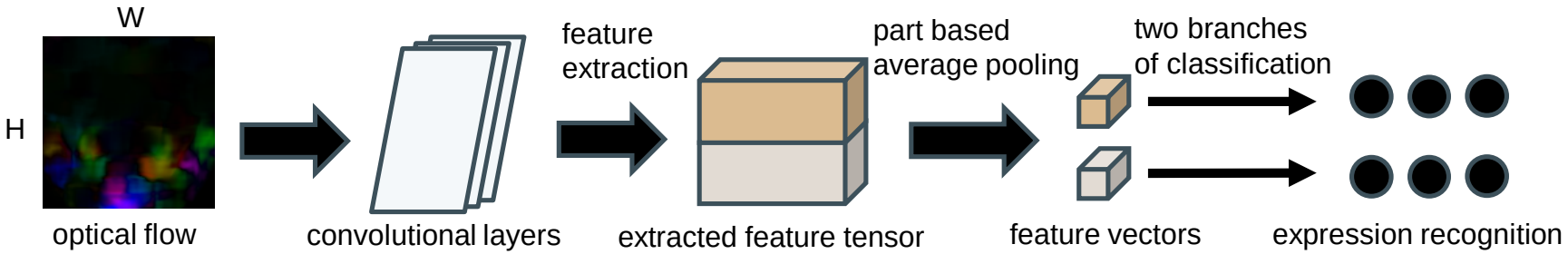
Challenges

- Short micro-expression duration
- Limited training data (Deep based methods suffer from this)

- Low intensity of facial muscle movements
- Domain gap between different datasets.

Methods

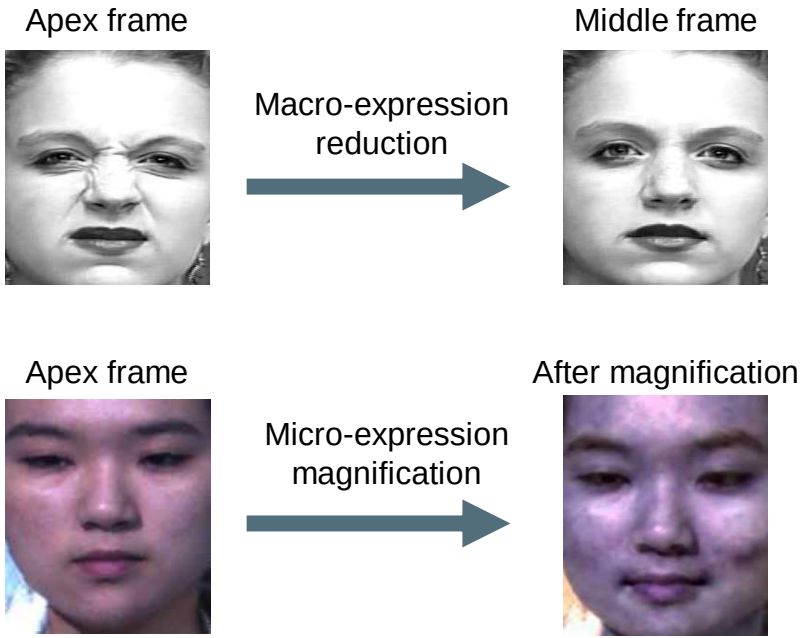
- Pre-processing
- Motion Feature Extraction
- Adversarial Domain Adaptation
- Part-Based Classification



Methods

- Supervised Domain Adaptation
  - Expression Magnification and Reduction (EMR)
    - Macro-expression reduction: the middle frame between the onset frame and the apex frame of the macro-expression video clip (CK+) is picked.
    - Micro-expression magnification: Eulerian Video Magnification (EVM) is used to magnify micro-expressions.

An example of EMR



Result

Method	Full		SMIC Part		CASME II Part		SAMM Part	
	UF1	UAR	UF1	UAR	UF1	UAR	UF1	UAR
LBP-TOP	0.5882	0.5785	0.2000	0.5280	0.7026	0.7429	0.3954	0.4102
Part + EMR	0.7663	0.7531	0.7001	0.6859	0.8615	0.8398	0.7180	0.6994
Part + Adversarial + EMR	0.7885	0.7824	0.7461	0.7530	0.8293	0.8209	0.7754	0.7152

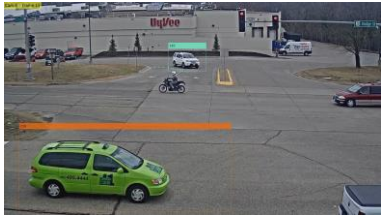
Vehicle Multi-Target Multi-Camera Tracking

Vehicle MTMCT

Camera 6

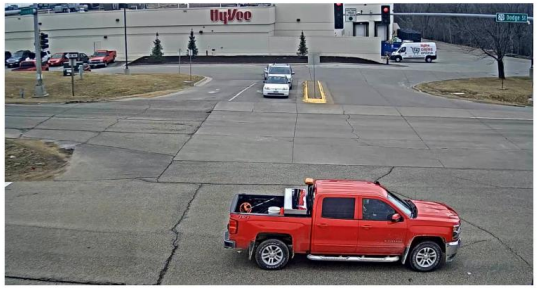
Camera 7

Camera 8

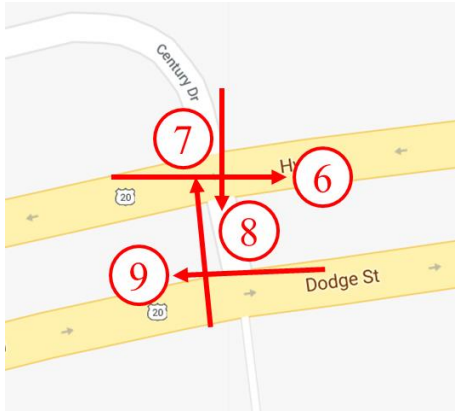


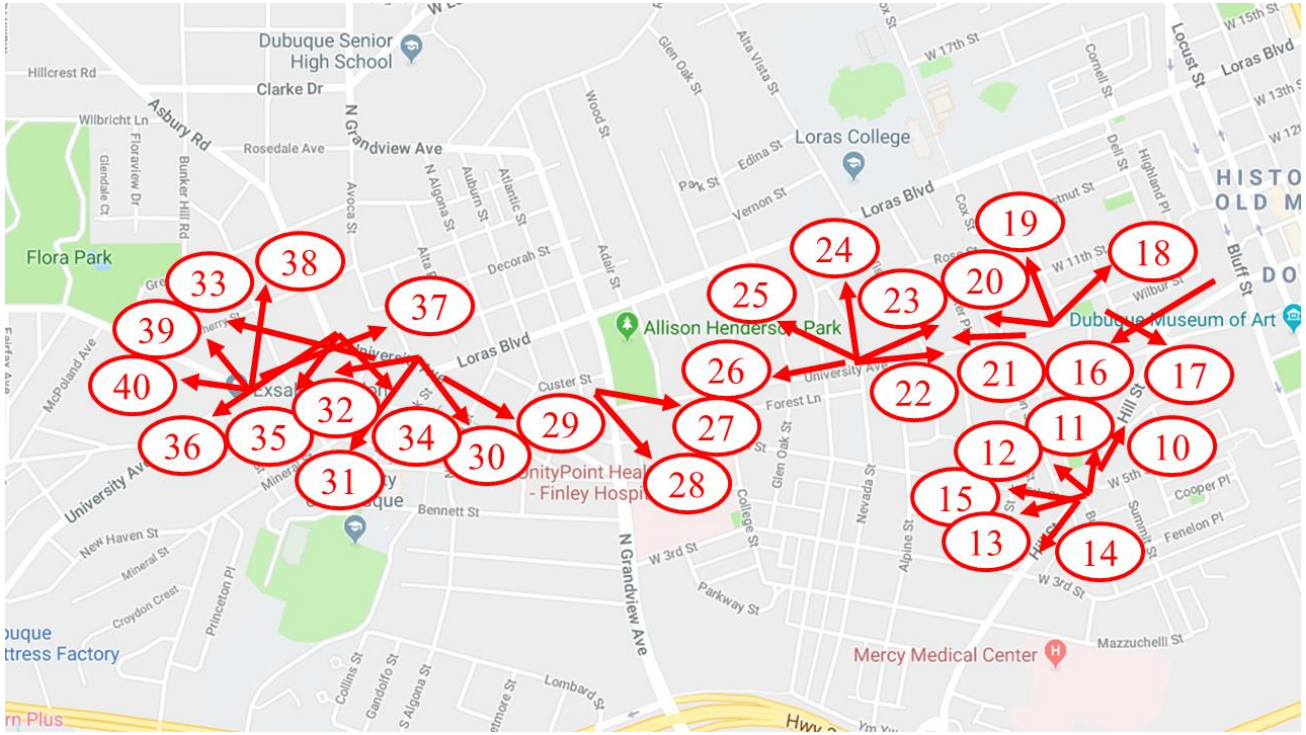
Camera 9

AI-City Dataset

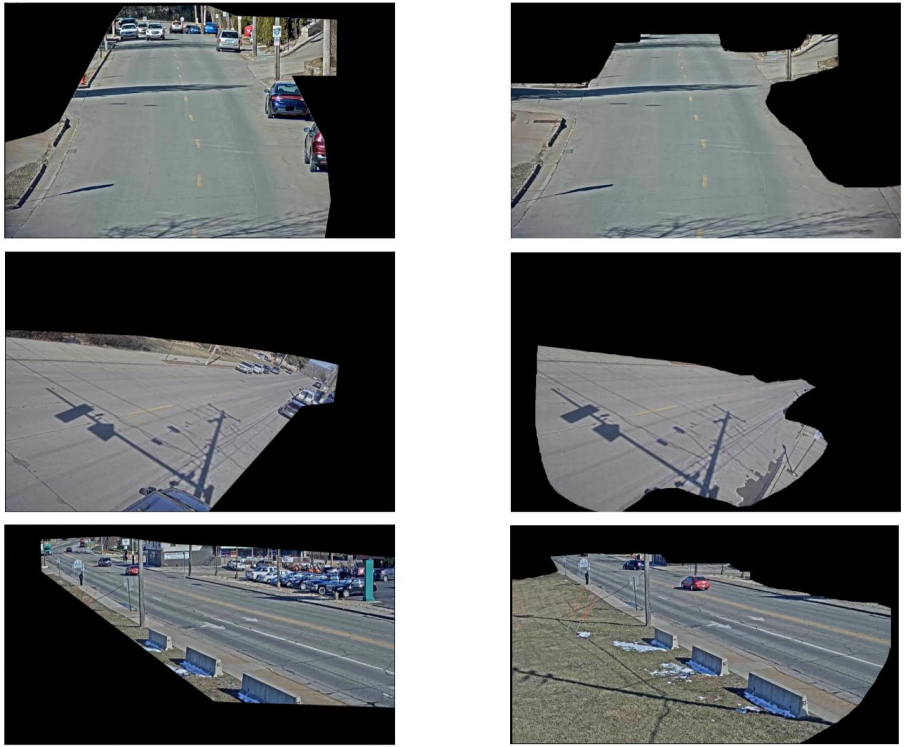


Challenges

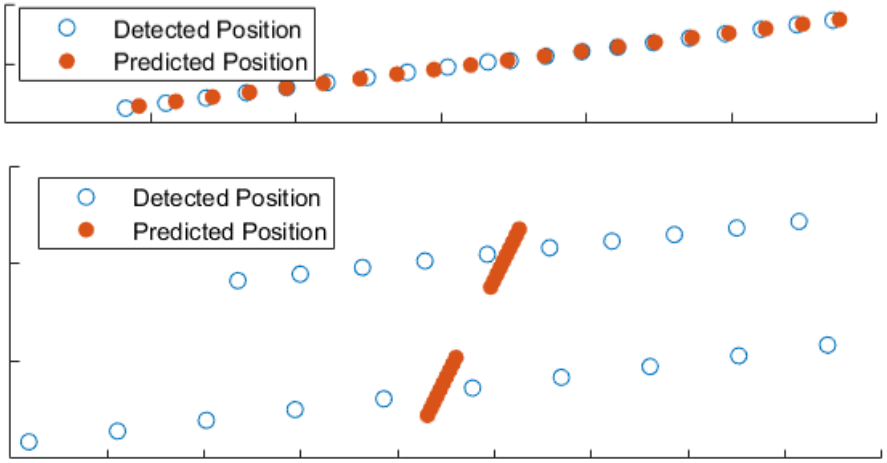

Century Dr
7
6
8
9
Dodge St
20
20


Dubuque Senior High School
Hillcrest Rd
Clarke Dr
Wilbricht Ln
Asbury Rd
Rosedale Ave
N Grandview Ave
Flora Park
Loras College
Loras Blvd
Allison Henderson Park
Dubuque Museum of Art
Decorah St
Custer St
Forest Ln
University Ave
Bennett St
New Haven St
Mineral St
Croydon Crest
Princeton Pl
W 3rd St
Parkway St
Mercy Medical Center
Mazzuchelli St
Cooper Pl
Fenelon Pl
W 5th St
W 11th St
W 17th St
W 15th St
Hill St
Bluff St
Locust St
Park St
Edina St
Vernon St
Wood St
Glen Oak St
Alta Vista St
Auburn St
Atlantic St
Adair St
Avoca St
Nevada St
Alpine St
College St
Lombard St
Collins St
Gandolfo St
Hwy
1
2
3
4
5
6
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
33
34
35
36
37
38
39
40

Refine ROI



Smooth Trajectories Association



Sub-scenario Division

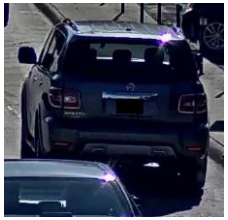
Result

Rank	Team ID	IDF Score
1	21	0.7059
2	49	0.6865
3	12	0.6653
4	53	0.6644
5	**97 (Ours)**	**0.6519**
6	59	0.5987
7	36	0.4924
8	107	0.4504
9	104	0.3369
10	52	0.2850

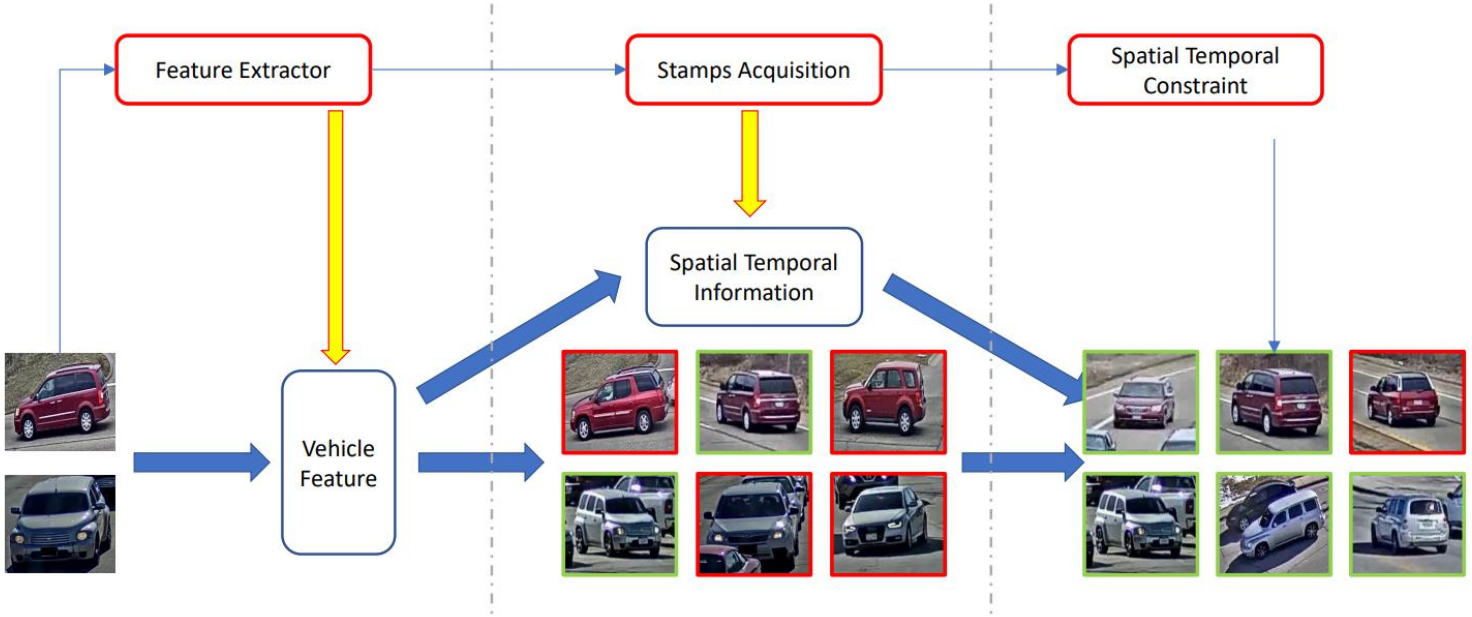
Vehicle Re-Identification

Vehicle Re-ID

Challenges



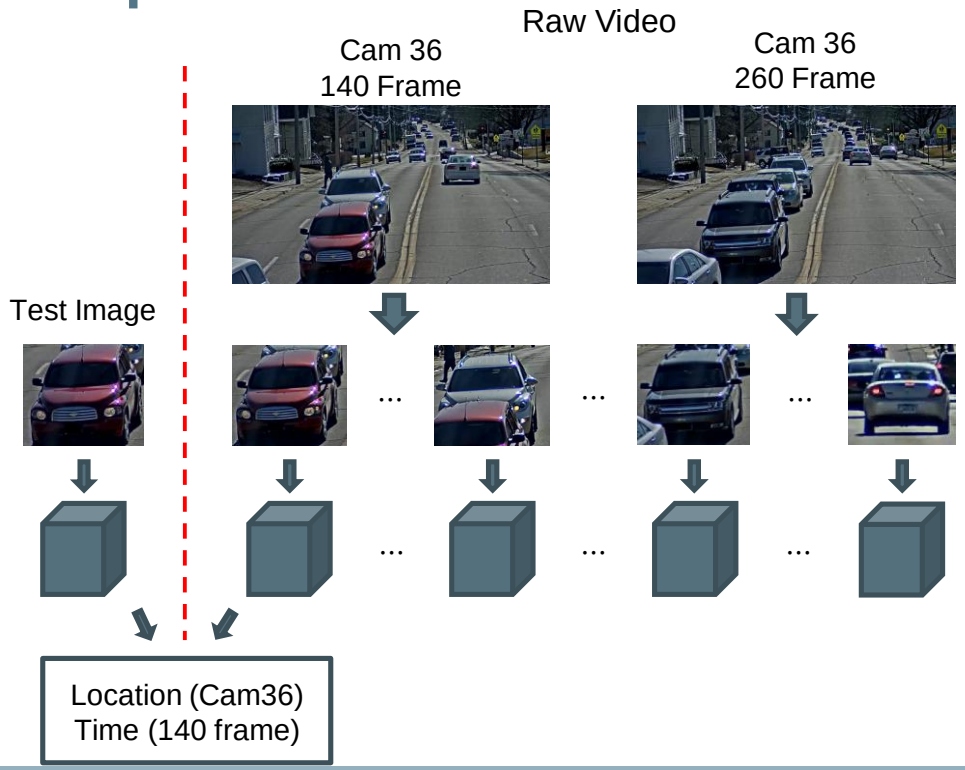
Vehicle Re-ID Framework



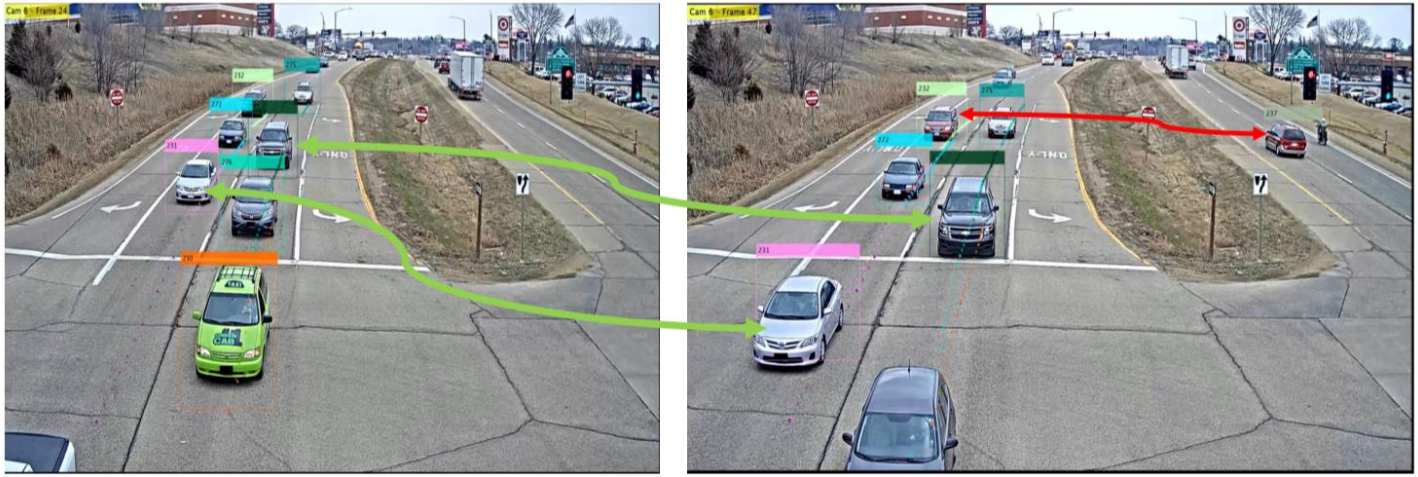
Feature Extractor

- Triplet loss and cross-entropy loss
- Feature ensemble
- Query expansion
- Temporal pooling for gallery

Stamps Acquisition



Spatial-Temporal Constraint



Frame 26 frame 47

Results

Methods	mAP	CMC_1	CMC_5	CMC_10
Baseline	0.3096	0.4677	0.4686	0.4734
Our Method	**0.7589**	**0.8916**	**0.8945**	**0.8973**

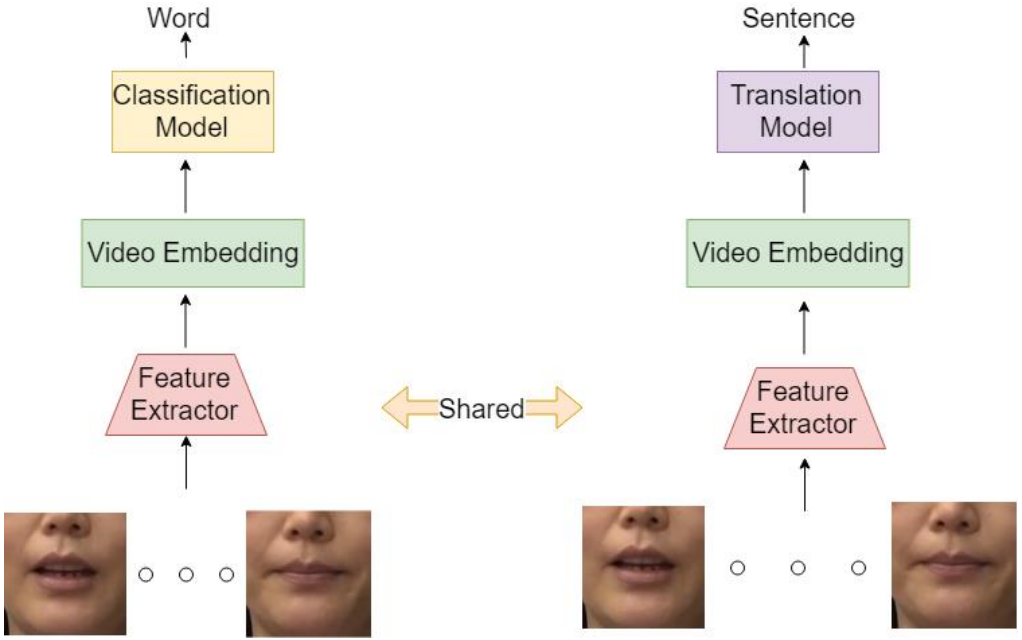
Lip Reading Keywords Spotting

Lip Reading KWS

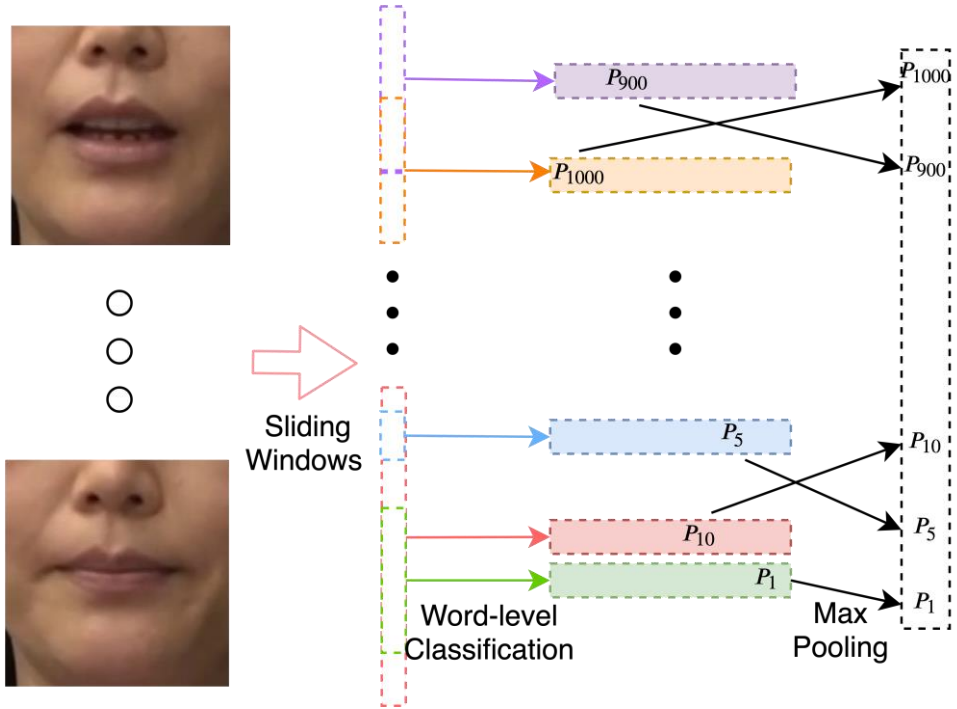
Exist or Not

Given Keyword

Challenges


Word
Classification Model
Video Embedding
Feature Extractor
Shared
Sentence
Translation Model
Video Embedding
Feature Extractor

Sliding Windows



Result

$$eIoU = \frac{X \cap Y}{X \cup Y}$$

Method	eIoU
Global Boundary	13.24%
Separate Boundary	**16.47%**
Top K	16.34%

Visual Navigation

Visual Navigation

Challenges

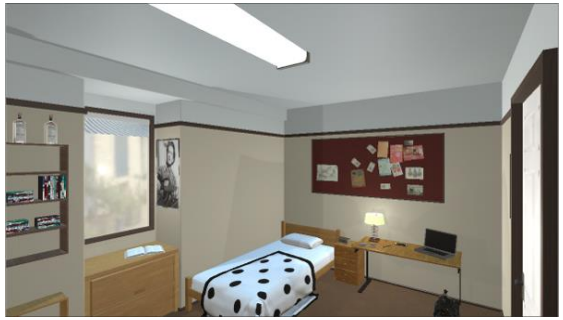
- Gap between target words and objects in the environment
- Exploring unseen objects effectively

AI2-thor Dataset

Kitchen

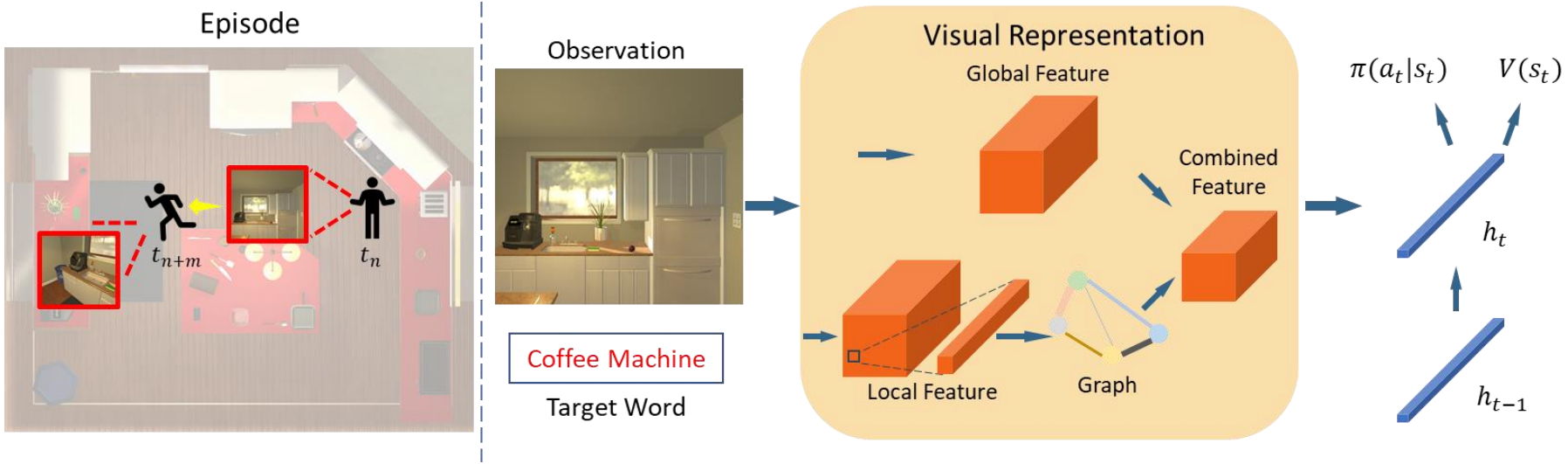
Living Room

Bedroom



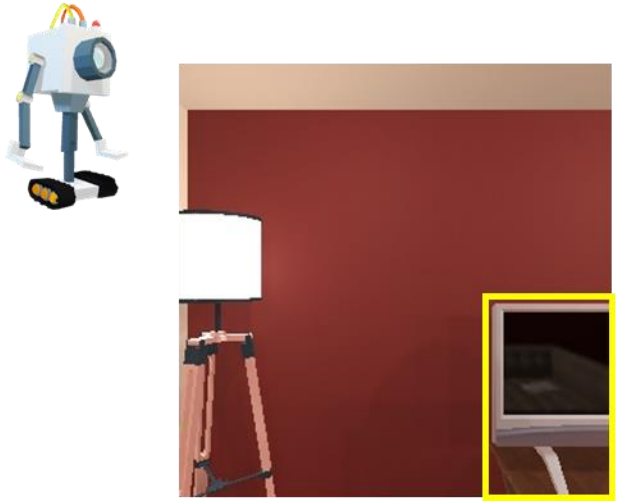
Bathroom

Overview


Episode
t_{n+m}
t_n
Observation
Coffee Machine
Target Word
Visual Representation
Global Feature
Combined Feature
Local Feature
Graph
$\pi(a_t|s_t)$
$V(s_t)$
h_t
h_{t-1}

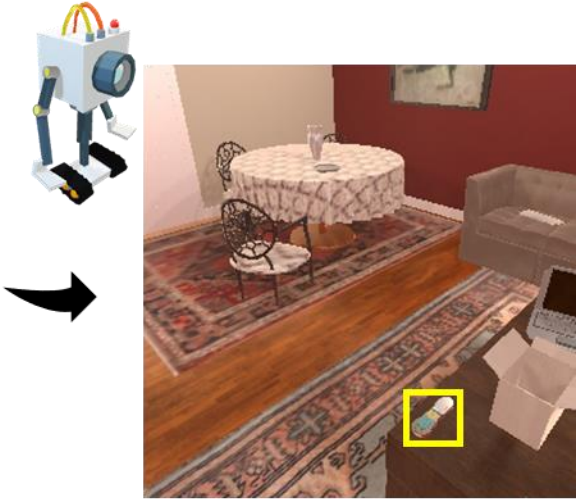
Results

I do not detect the **remote control**, But I detect one **TV** here.



Remote control should appear around the **TV**.

Sure, I detected the **remote control**.



Results

Method	Success Rate(%)	Success Weighted by Path Length (SPL)
Word Embedding + A3C	35.1	0.15
Global/Local Feature + A3C	82.3	0.51
Global/Local Feature + Graph + A3C	88.3	0.58

Future Work

- Develop the visual navigation system in multi-room scene
- Build instance-level system
- Evaluate the agent in the real-world situation
- Try something about synthetic data generation

Thanks!

Australian National University

Bidirectional Type Checking for Modal Types

Joseph Meltzer

Supervisor: Ranald Clouston

Contribution

- Type checker for modal type systems
  - Bidirectional type checking
  - Algorithm and Haskell implementation
  - Extension to dependent types

Contents

- Part 1: Introduction
  - The Curry-Howard Correspondence
  - Modal type systems
  - Motivation for modal type checking
- Part 2: Modal type checking
  - Bidirectional type checking
  - Modal error messages
  - Implementation
- Part 3: Dependent modal type checking
  - Dependent types
  - Evaluation and computation

Curry-Howard Correspondence

- Propositions = Types

$$\frac{A \quad B}{A \wedge B} \qquad \frac{x : A \quad y : B}{(x, y) : A \times B}$$

$$\frac{A \Rightarrow B \quad A}{B} \qquad \frac{f : A \to B \quad x : A}{f\ x : B}$$

$$\frac{X, A \vdash B}{X \vdash A \Rightarrow B} \qquad \frac{X, x : A \vdash e : B}{X \vdash \lambda x.\, e : A \to B}$$

Curry-Howard Correspondence

- Propositions = Types
- A proposition is true if and only if there exists a *term* of the corresponding type

$$\frac{A \quad B}{A \wedge B} \qquad \frac{x : A \quad y : B}{(x, y) : A \times B}$$

$$\frac{A \Rightarrow B \quad A}{B} \qquad \frac{f : A \to B \quad x : A}{f\ x : B}$$

$$\frac{X, A \vdash B}{X \vdash A \Rightarrow B} \qquad \frac{X, x : A \vdash e : B}{X \vdash \lambda x.\, e : A \to B}$$

Curry-Howard Correspondence

- Propositions = Types
- A proposition is true if and only if there exists a *term* of the corresponding type
- The *term* is a proof that the proposition is true

$$\frac{A \quad B}{A \wedge B} \qquad \frac{x : A \quad y : B}{(x, y) : A \times B}$$

$$\frac{A \Rightarrow B \quad A}{B} \qquad \frac{f : A \to B \quad x : A}{f\ x : B}$$

$$\frac{X, A \vdash B}{X \vdash A \Rightarrow B} \qquad \frac{X, x : A \vdash e : B}{X \vdash \lambda x.\, e : A \to B}$$

Curry-Howard Correspondence

Logic	Type theory
Propositions	Types
Proofs	Programs
Intuitionistic logic	Simply-typed λ-calculus
First-order logic	Dependent types
Second-order logic	Polymorphic types
Proof checking	Type checking

Modal Logic

- Extend any logic with modalities
- Temporal logic, concurrency, belief vs. truth
- $\Box A$ means that A is necessarily/always/believed-to-be true
- Statement about a generic ‘world’

Different modal logics

All (normal) modal logics start with Axiom K:

$$\Box\,(A \to B) \to \Box\,A \to \Box\,B$$

Intuitionistic K	Intuitionistic S4	Intuitionistic R
No extra axioms	T: $\Box A \to A$ 4: $\Box A \to \Box(\Box A)$	R: $A \to \Box A$

Modal type systems

- Propositions → types
- Modal logic → modal type system
- New constructors `open` and `shut`
- New and altered typing rules to reflect new calculus
- Variables can be shut behind locks

Modal types for programming

- IS4 = Comonads
  - streams, FFT, cellular automata
- IR = Applicative functors
  - lists, optionals, *IO*
- Haskell uses typeclasses (interfaces) to express the rules of these structures

Modal types for programming

```
class Applicative f where
   fmap  :: (a -> b) -> f a -> f b
   pure  :: a -> f a
   apply :: f (a -> b) -> f a -> f b
```

$R\colon A \to \Box A$

$\mathrm{K}\colon \Box (A \to B) \to \Box A \to \Box B$

- Haskell typeclasses have extra constraints

```
fmap f x = apply (pure f) x
```

- Haskell can't enforce these at compile time
- Dependent types can do this but are complicated
- Modal types give these properties for free

Modal types for proof verification

- Proof verification is difficult
- Especially when proof logic does not match proof language
  - Coq and Agda based on intuitionistic logic
- Modal logic proofs require declaring axioms
- Theorem proving is best done with natural deduction
- Proving with axioms is tedious (Hilbert-style proofs)

Modal types for proof verification

- Type checking is proof checking
- Type checker for modal types = proof checker for modal logic
- Modal typing rules = natural deduction rules for modal logic
- Modal proof checker allows natural deduction proofs instead of Hilbert-style proofs

Modal type checking

- Bidirectional type checking technique
  - Two directions: inference and checking
  - Type inference to reduce annotations
  - Not complete, but annotations can be useful
  - Good error messages
  - Extendable to advanced type systems
- I implement this for modal λ-calculi

Modal type checking

- Bidirectional typing rules for the modal elements of the modal λ-calculi
- Rules are sound with respect to existing modal λ-calculi
- I devised an algorithm to perform bidirectional type checking on modal λ-calculi

Modal error messages

- Locked variable (IK and IS4)
- Not a box
- Illegal open (IK and IR)
- Failed open (IS4 and IR)

Implementation

- Haskell implementation for modal λ-calculi
  - Bidirectional type checking
  - Basic term computation (lazy evaluation)
  - Useful error messages
- Correctness testing for axioms and properties

Extension: dependent types

- Dependent types contain values
  - Vector of fixed length, e.g. `Vec 5` or `Vec 12`
  - `[1, 2, 3, 4, 5] :: Vec 5`
  - `concat :: Vec n → Vec m → Vec (n+m)`
- Equality is important and complicated
  - `Vec 5 = Vec (2+3)`
  - Type checking requires equality of terms

Why dependent types?

- Dependent types = first-order logic
- Can enforce almost any constraint at compile-time
- Too complicated for general programming
- Essential for proof checkers
  - Nearly all proofs use first-order logic

Evaluating modal types

- Need to check equality of types A and B
  - Evaluate A and B to their normal forms and compare exact syntactic equality
- Evaluating types means evaluating terms
- Evaluate new constructors `open` and `shut`
  - `open shut t = t` (beta equivalence)
  - `shut open t = t` (eta equivalence)

Eta (η) expansion

- η-expansion for functions:
  - If `f` is a function, then `λx.(f x) = f`
  - Best normal forms achieved by *expanding* `f` to `λx.(f x)`
- η-expansion for modal types:
  - If `t` has a boxed type, then `shut open t = t`
  - Must expand `t` to `shut open t`

Dependent modal types

$$\Box\,(A \to B) \to \Box\,A \to \Box\,B$$

- Axiom K can be expressed in a stronger form with dependent types
- Axiom K also has equivalent non-dependent forms using conjunction
- I formulate stronger dependent versions of these equivalent forms.

Contributions

- For the modal λ-calculi IK, IS4, IR:
  - Algorithmic bidirectional type checking
  - Computation by lazy evaluation
  - Error message theory for modal types
  - Haskell implementation of the above
  - Correctness testing of implementation

Contributions

- For the dependent modal λ-calculi K, S4, and R
  - As before, but with full evaluation (normal order)
  - Theory of η-expansion for modal types
  - Dependently typed forms of Axiom K's equivalent expression using conjunctions

Future work

- Other modal logics
  - Wider improvement to formal verification
  - Correctness for more structures, e.g. monads
- Advanced language features
  - Polymorphism, natural numbers, vectors
- Multiple/mixed modalities
  - Proof check systems with multiple modalities
  - Allow more than one comonad or applicative functor

Future work

- Proof of correctness
  - Dependent λ-calculi for S4 and R require proof of good computational properties
- Formal verification
  - Verify that the algorithm and implementation correctly perform the bidirectional type checking rules
  - Important for incorporating into real proof languages

Thank you