ActionNet: Vision-based Workflow Action Recognition From Programming Screencasts

Dehai Zhao, Zhenchang Xing, Chunyang Chen, Xin Xia, Guoqiang Li

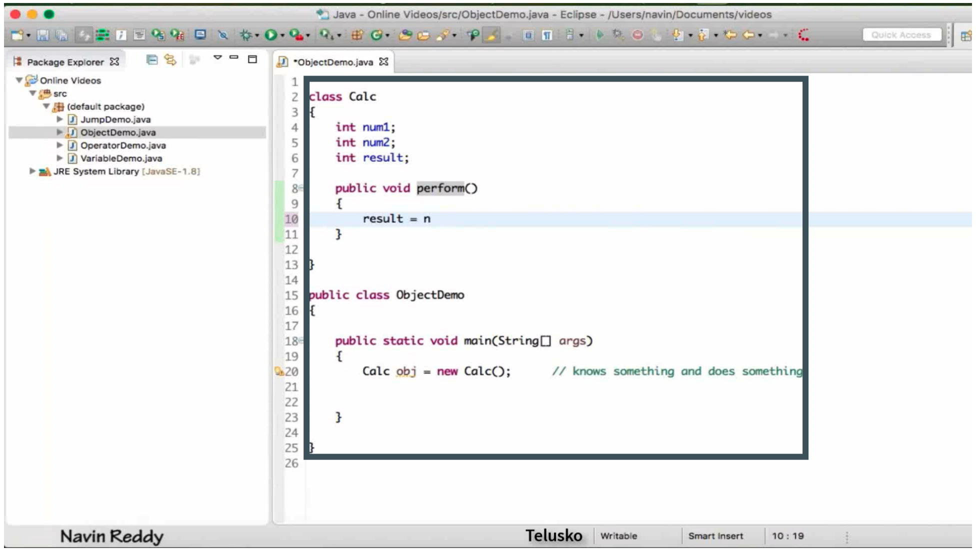
1 March 2020

Research School of Computer Science
College of Engineering & Computer Science
The Australian National University
Canberra, Australia

Content

- Motivation
- Problem Statement
- Approach
- Evaluation
- Results

Motivation

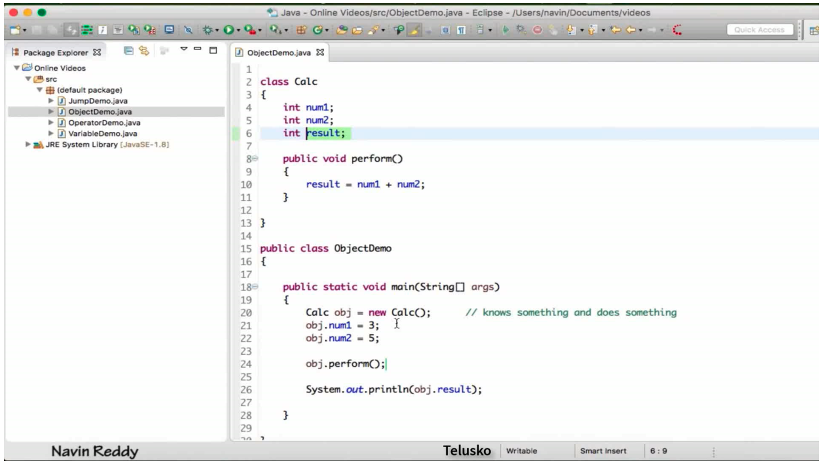


```

class Calc
{
    int num1;
    int num2;
    int result;

    public void perform()
    {
        result = n
    }

}

public class ObjectDemo
{

    public static void main(String[] args)
    {
        Calc obj = new Calc();      // knows something and does something


    }

}

```

- Feature location
- Debugging
- Program comprehension
- Tool design
- Distributed programming

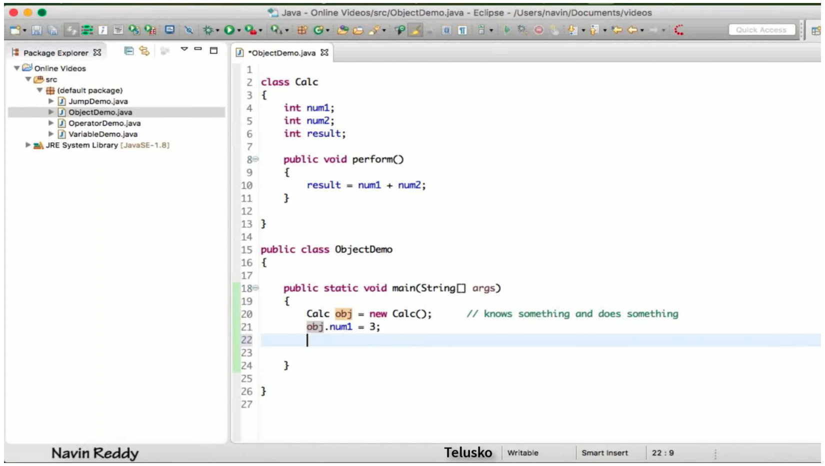
```
class calc
{
    int num1;
    int num2;
    int result;
    …..
```

Motivation



```java

class Calc
{
    int num1;
    int num2;
    int result;

    public void perform()
    {
        result = num1 + num2;
    }

}

public class ObjectDemo
{

    public static void main(String[] args)
    {
        Calc obj = new Calc();      // knows something and does something
        obj.num1 = 3;
        obj.num2 = 5;

        obj.perform();

        System.out.println(obj.result);

    }

```

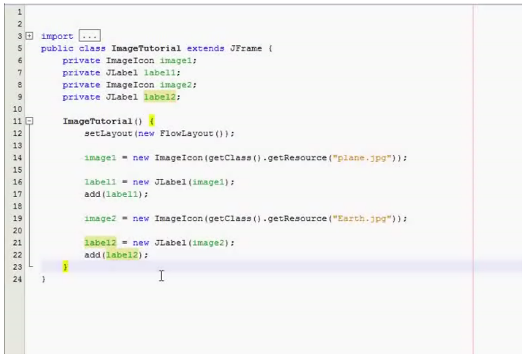
Selecting



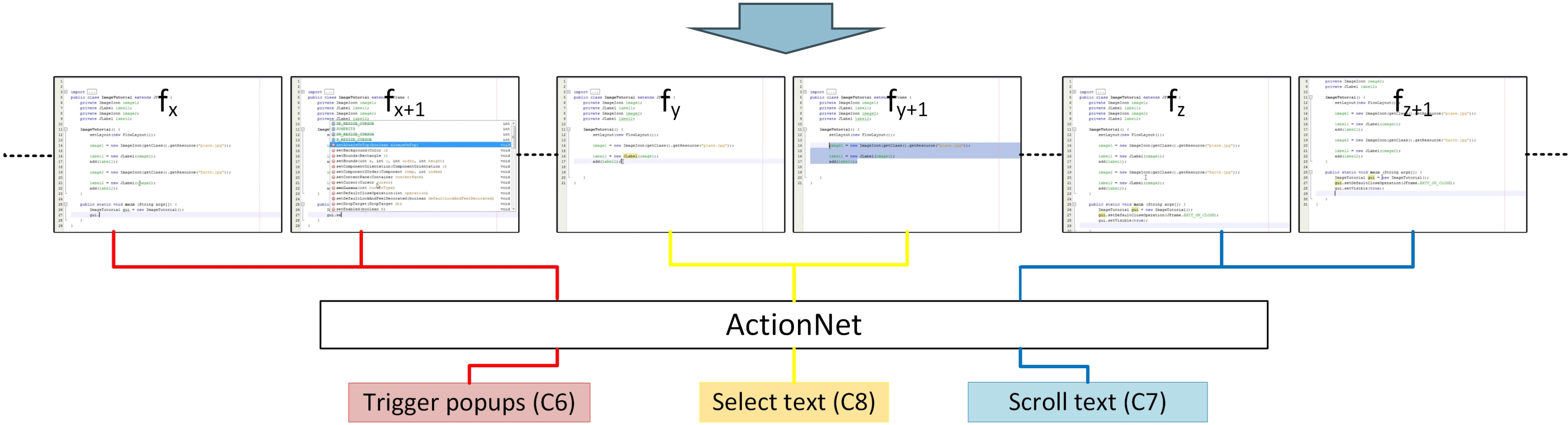
```java

class Calc
{
    int num1;
    int num2;
    int result;

    public void perform()
    {
        result = num1 + num2;
    }

}

public class ObjectDemo
{

    public static void main(String[] args)
    {
        Calc obj = new Calc();      // knows something and does something
        obj.num1 = 3;


    }

}

```

Popup window

Problem Statement



```
import ...
public class ImageTutorial extends JFrame {
    private ImageIcon image1;
    private JLabel label1;
    private ImageIcon image2;
    private JLabel label2;

    ImageTutorial() {
        setLayout(new FlowLayout());

        image1 = new ImageIcon(getClass().getResource("plane.jpg"));

        label1 = new JLabel(image1);
        add(label1);

        image2 = new ImageIcon(getClass().getResource("Earth.jpg"));

        label2 = new JLabel(image2);
        add(label2);
    }
}
```



f_x f_{x+1} f_y f_{y+1} f_z f_{z+1}

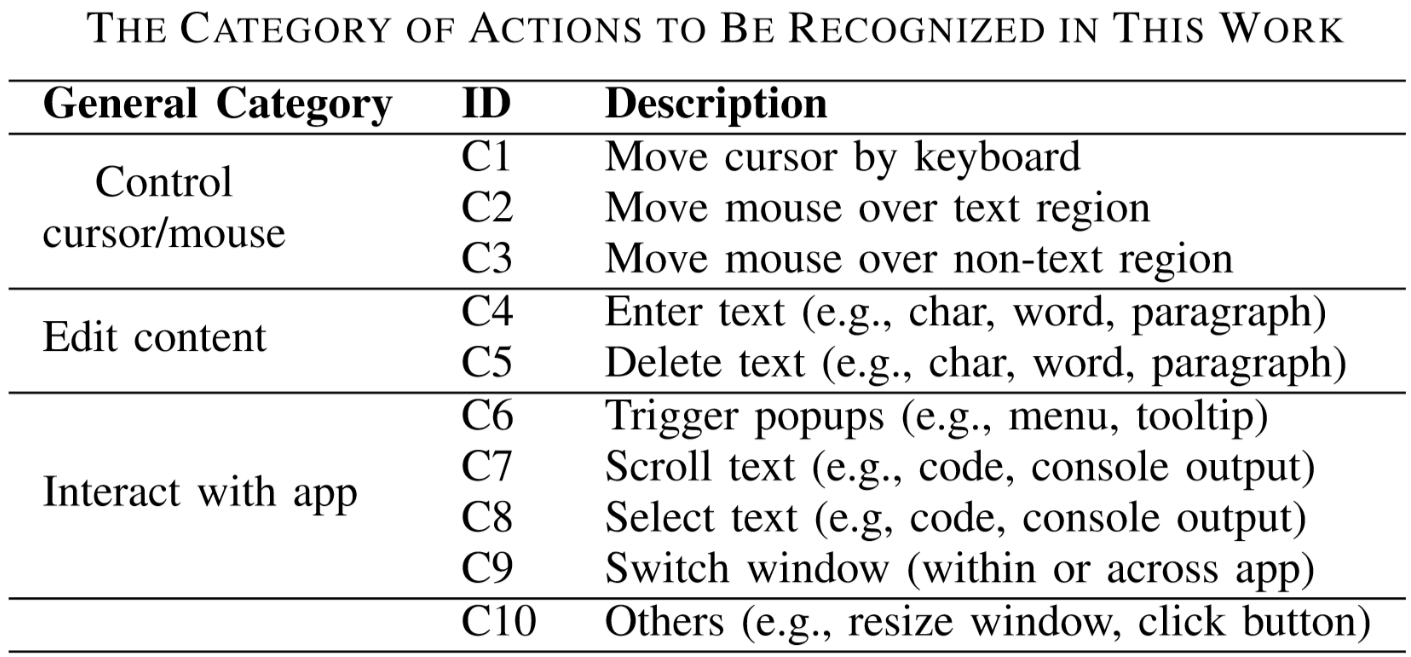
ActionNet

Trigger popups (C6)

Select text (C8)

Scroll text (C7)

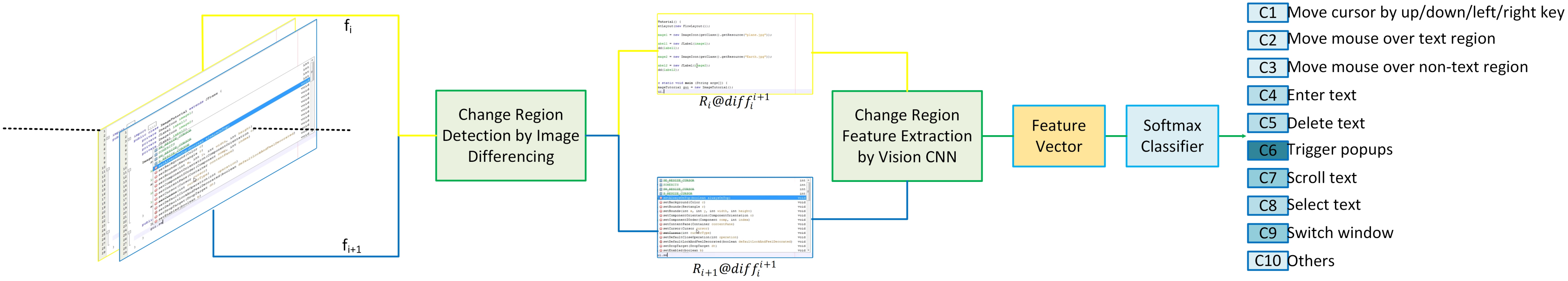
Problem Statement



THE CATEGORY OF ACTIONS TO BE RECOGNIZED IN THIS WORK

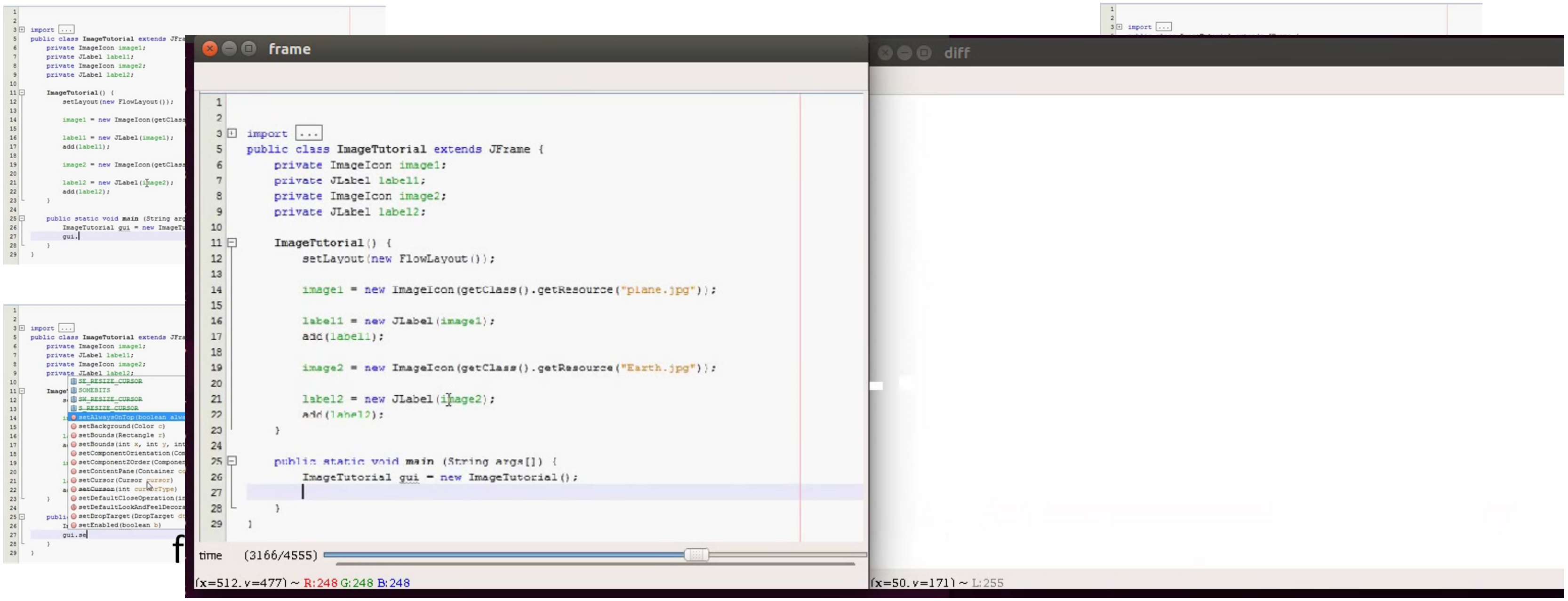
General Category	ID	Description
Control cursor/mouse	C1	Move cursor by keyboard
	C2	Move mouse over text region
	C3	Move mouse over non-text region
Edit content	C4	Enter text (e.g., char, word, paragraph)
	C5	Delete text (e.g., char, word, paragraph)
Interact with app	C6	Trigger popups (e.g., menu, tooltip)
	C7	Scroll text (e.g., code, console output)
	C8	Select text (e.g, code, console output)
	C9	Switch window (within or across app)
	C10	Others (e.g., resize window, click button)

Approach


f_i
f_{i+1}
Change Region Detection by Image Differencing
$R_i@diff_i^{i+1}$
$R_{i+1}@diff_i^{i+1}$
Change Region Feature Extraction by Vision CNN
Feature Vector
Softmax Classifier
C1 Move cursor by up/down/left/right key
C2 Move mouse over text region
C3 Move mouse over non-text region
C4 Enter text
C5 Delete text
C6 Trigger popups
C7 Scroll text
C8 Select text
C9 Switch window
C10 Others

An Overview of the Main Steps of Our ActionNet

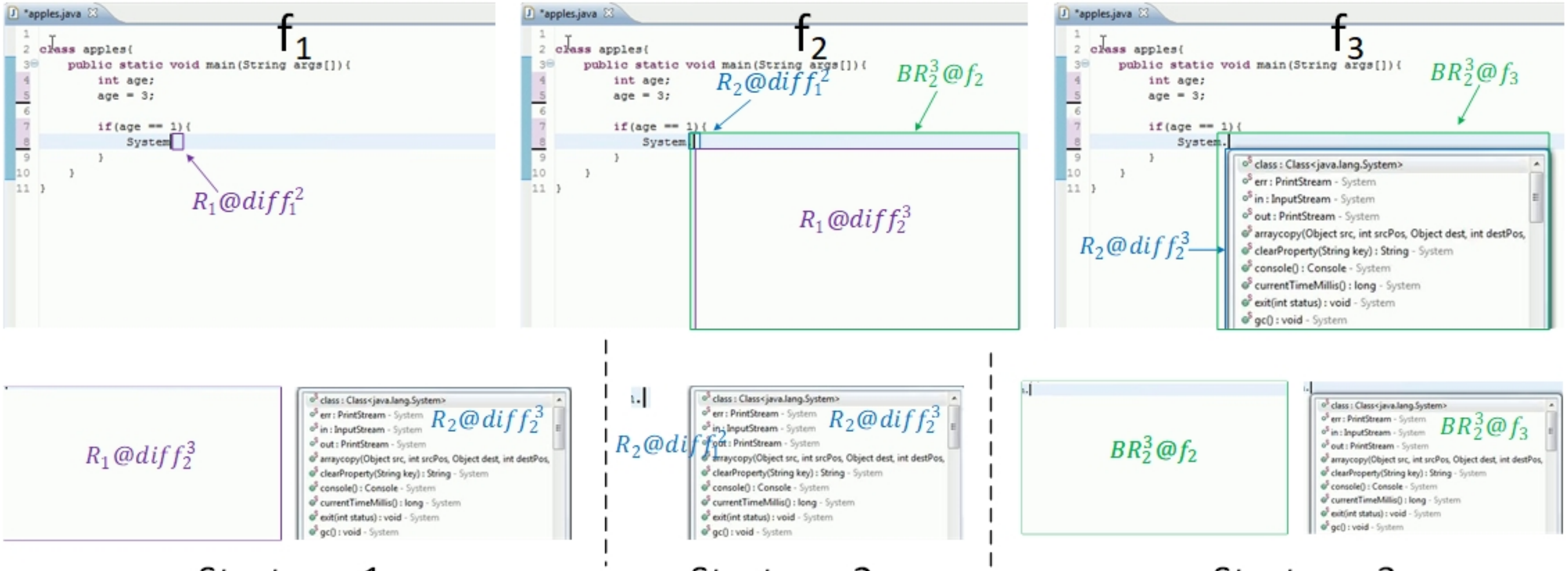
Approach



```
import ...
public class ImageTutorial extends JFrame {
    private ImageIcon image1;
    private JLabel label1;
    private ImageIcon image2;
    private JLabel label2;

    ImageTutorial() {
        setLayout(new FlowLayout());

        image1 = new ImageIcon(getClass().getResource("plane.jpg"));

        label1 = new JLabel(image1);
        add(label1);

        image2 = new ImageIcon(getClass().getResource("Earth.jpg"));

        label2 = new JLabel(image2);
        add(label2);
    }

    public static void main (String args[]) {
        ImageTutorial gui = new ImageTutorial();

    }
}
```

Steps to Detect Changes Regions in Between fi and fi+1

Approach



*apples.java

```
class apples{
    public static void main(String args[]){
        int age;
        age = 3;

        if(age == 1){
            System.
        }
    }
}
```

```
class : Class<java.lang.System>
err : PrintStream - System
in : InputStream - System
out : PrintStream - System
arraycopy(Object src, int srcPos, Object dest, int destPos,
clearProperty(String key) : String - System
console() : Console - System
currentTimeMillis() : long - System
exit(int status) : void - System
gc() : void - System
```

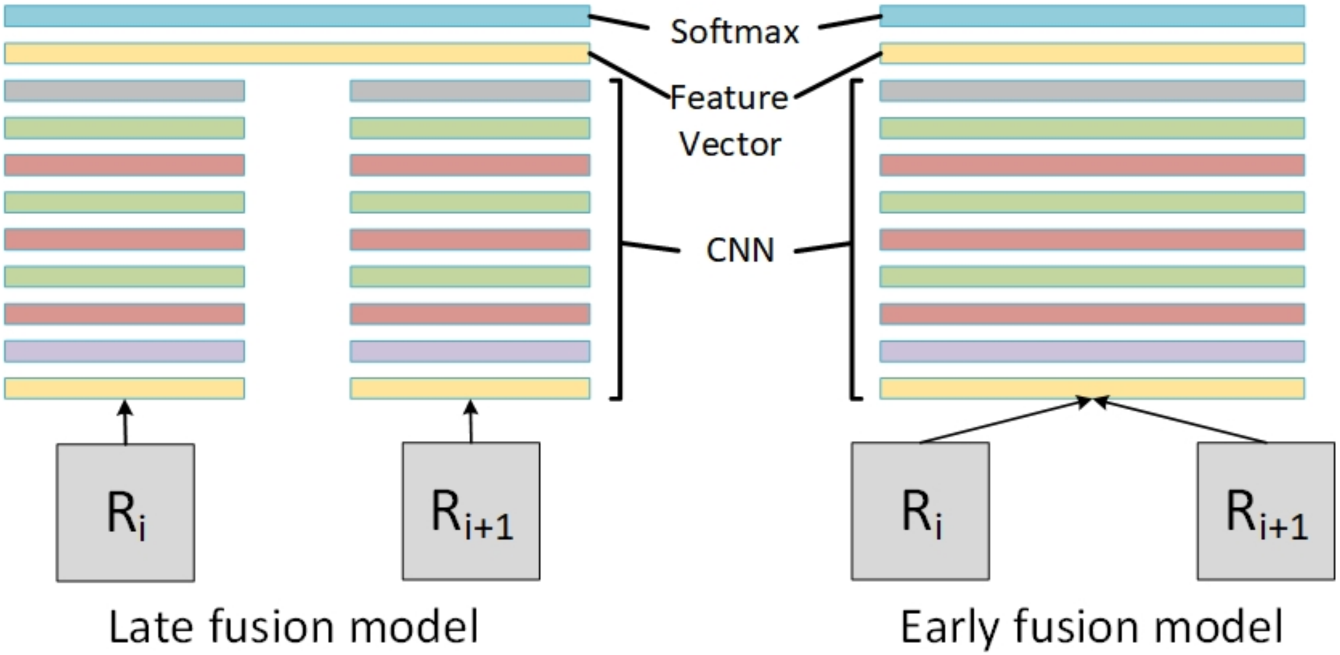
Strategy-1

Strategy-2

Strategy-3

Illustration of Three Strategies for Input Change Regions

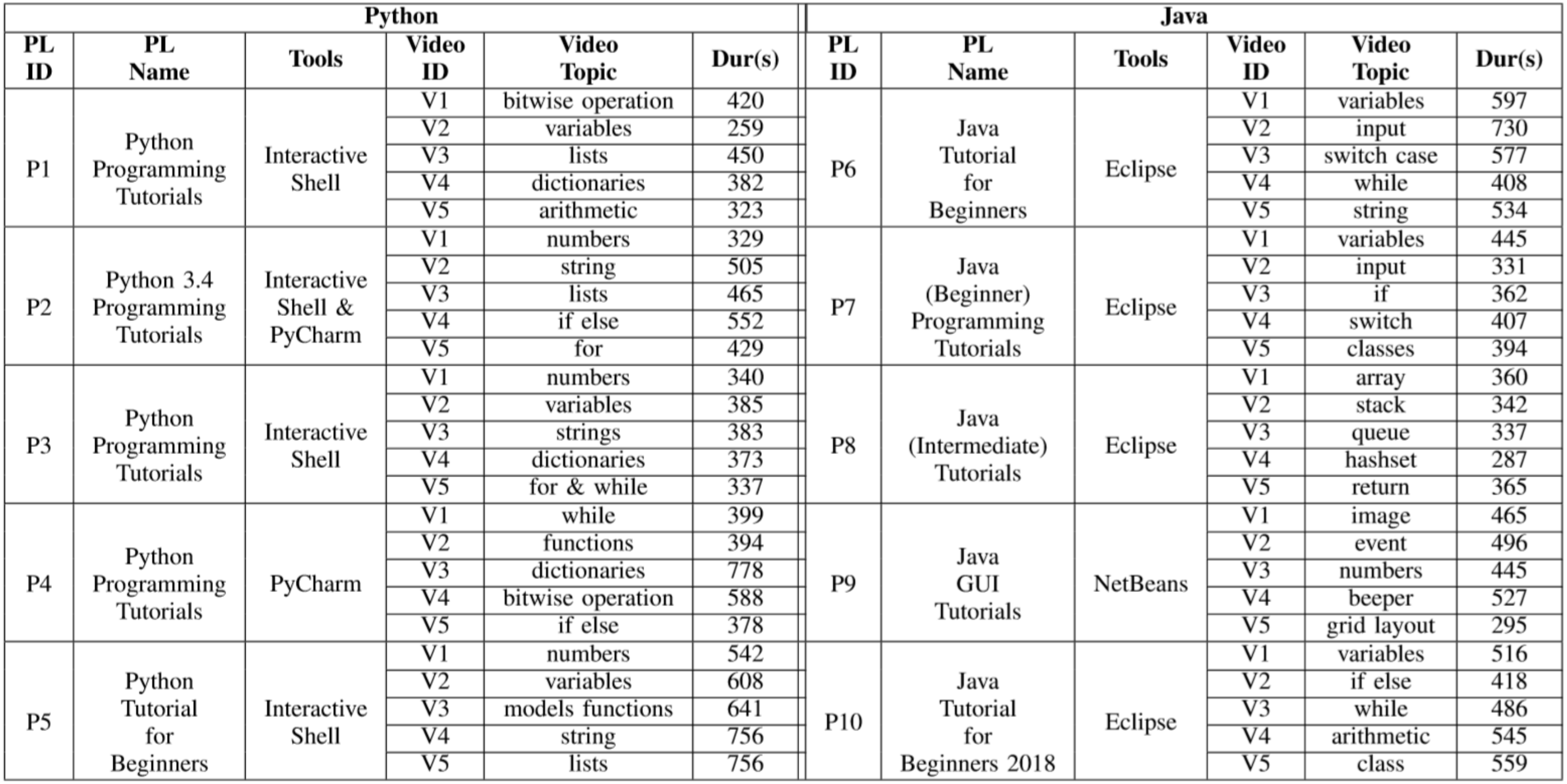
Approach



Early Fusion versus Late Fusion Architecture

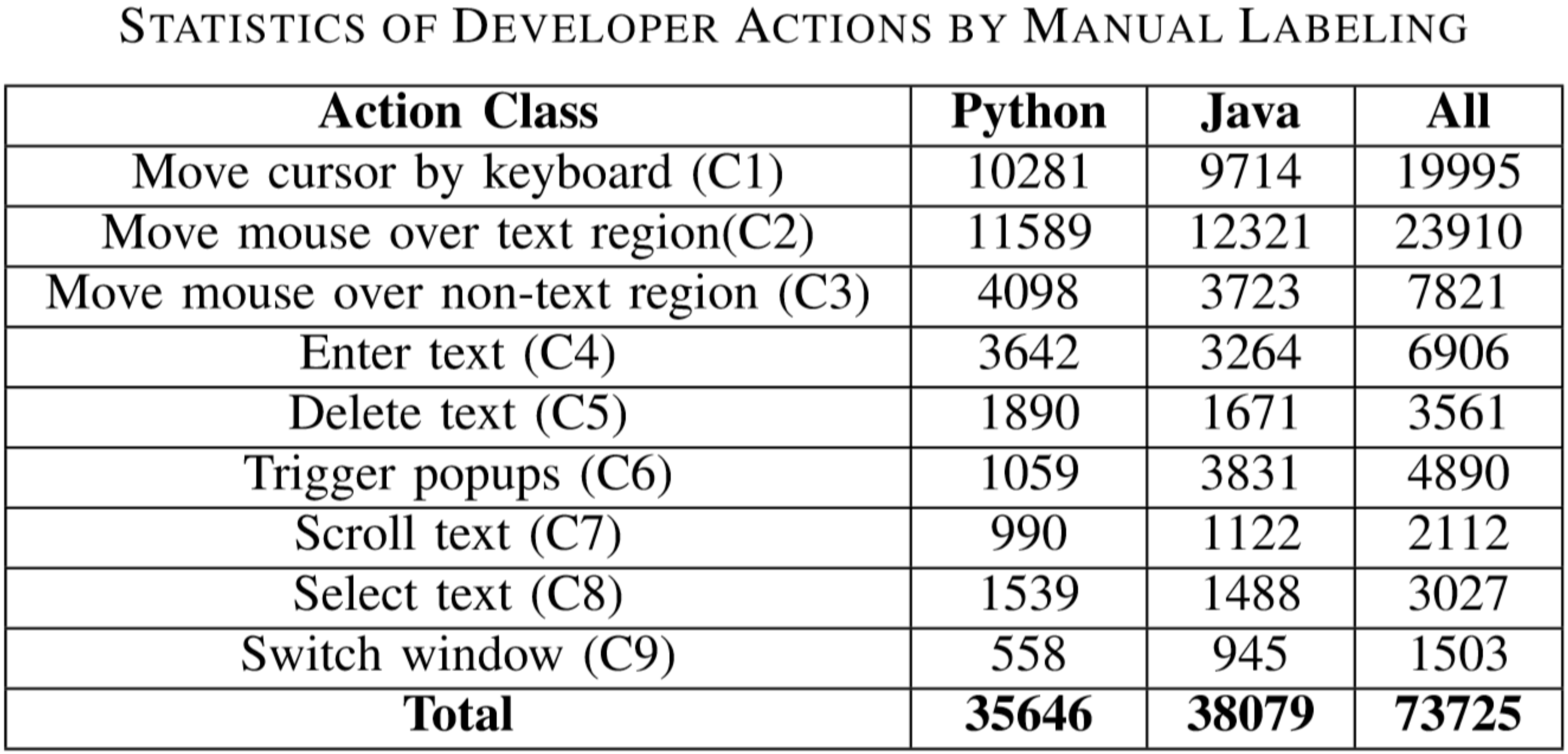
Evaluation

The Dataset of Programming Screencasts Crawled from YouTube



Python						Java					
PL ID	**PL Name**	**Tools**	**Video ID**	**Video Topic**	**Dur(s)**	**PL ID**	**PL Name**	**Tools**	**Video ID**	**Video Topic**	**Dur(s)**
P1	Python Programming Tutorials	Interactive Shell	V1	bitwise operation	420	P6	Java Tutorial for Beginners	Eclipse	V1	variables	597
			V2	variables	259				V2	input	730
			V3	lists	450				V3	switch case	577
			V4	dictionaries	382				V4	while	408
			V5	arithmetic	323				V5	string	534
P2	Python 3.4 Programming Tutorials	Interactive Shell & PyCharm	V1	numbers	329	P7	Java (Beginner) Programming Tutorials	Eclipse	V1	variables	445
			V2	string	505				V2	input	331
			V3	lists	465				V3	if	362
			V4	if else	552				V4	switch	407
			V5	for	429				V5	classes	394
P3	Python Programming Tutorials	Interactive Shell	V1	numbers	340	P8	Java (Intermediate) Tutorials	Eclipse	V1	array	360
			V2	variables	385				V2	stack	342
			V3	strings	383				V3	queue	337
			V4	dictionaries	373				V4	hashset	287
			V5	for & while	337				V5	return	365
P4	Python Programming Tutorials	PyCharm	V1	while	399	P9	Java GUI Tutorials	NetBeans	V1	image	465
			V2	functions	394				V2	event	496
			V3	dictionaries	778				V3	numbers	445
			V4	bitwise operation	588				V4	beeper	527
			V5	if else	378				V5	grid layout	295
P5	Python Tutorial for Beginners	Interactive Shell	V1	numbers	542	P10	Java Tutorial for Beginners 2018	Eclipse	V1	variables	516
			V2	variables	608				V2	if else	418
			V3	models functions	641				V3	while	486
			V4	string	756				V4	arithmetic	545
			V5	lists	756				V5	class	559

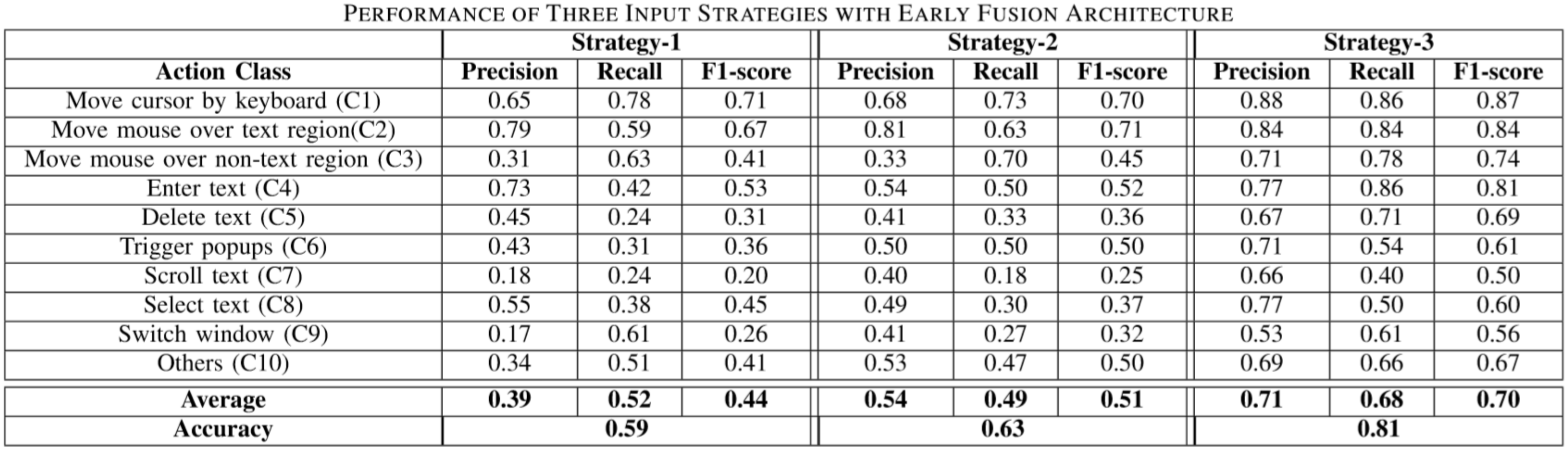
Evaluation



STATISTICS OF DEVELOPER ACTIONS BY MANUAL LABELING

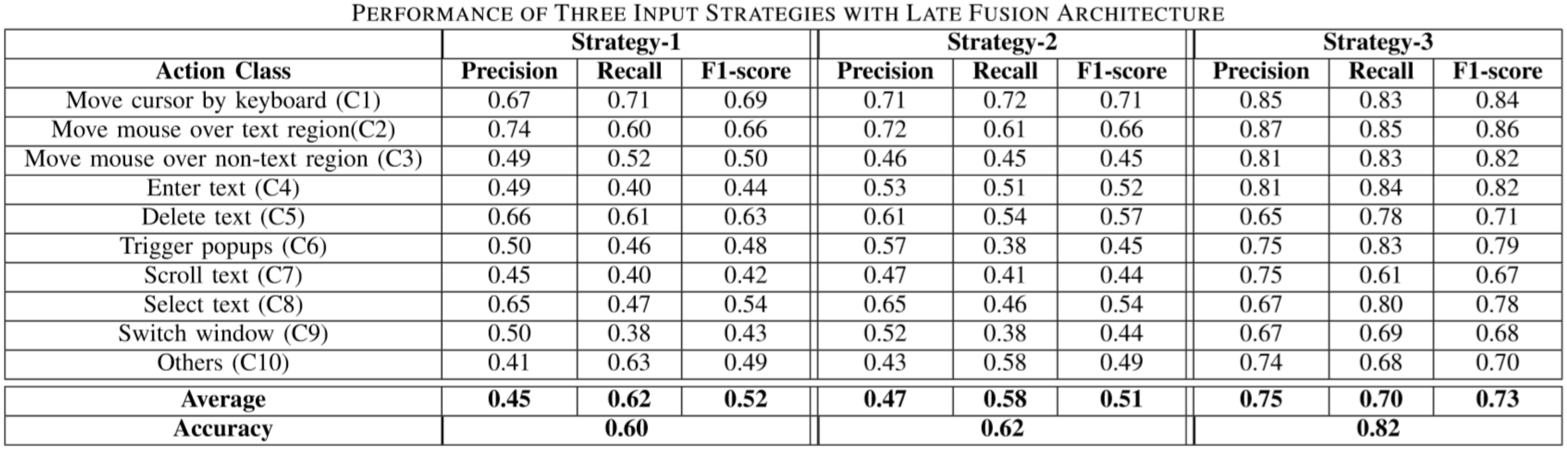
Action Class	Python	Java	All
Move cursor by keyboard (C1)	10281	9714	19995
Move mouse over text region(C2)	11589	12321	23910
Move mouse over non-text region (C3)	4098	3723	7821
Enter text (C4)	3642	3264	6906
Delete text (C5)	1890	1671	3561
Trigger popups (C6)	1059	3831	4890
Scroll text (C7)	990	1122	2112
Select text (C8)	1539	1488	3027
Switch window (C9)	558	945	1503
Total	**35646**	**38079**	**73725**

Evaluation



PERFORMANCE OF THREE INPUT STRATEGIES WITH EARLY FUSION ARCHITECTURE

	Strategy-1			Strategy-2			Strategy-3		
Action Class	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Move cursor by keyboard (C1)	0.65	0.78	0.71	0.68	0.73	0.70	0.88	0.86	0.87
Move mouse over text region(C2)	0.79	0.59	0.67	0.81	0.63	0.71	0.84	0.84	0.84
Move mouse over non-text region (C3)	0.31	0.63	0.41	0.33	0.70	0.45	0.71	0.78	0.74
Enter text (C4)	0.73	0.42	0.53	0.54	0.50	0.52	0.77	0.86	0.81
Delete text (C5)	0.45	0.24	0.31	0.41	0.33	0.36	0.67	0.71	0.69
Trigger popups (C6)	0.43	0.31	0.36	0.50	0.50	0.50	0.71	0.54	0.61
Scroll text (C7)	0.18	0.24	0.20	0.40	0.18	0.25	0.66	0.40	0.50
Select text (C8)	0.55	0.38	0.45	0.49	0.30	0.37	0.77	0.50	0.60
Switch window (C9)	0.17	0.61	0.26	0.41	0.27	0.32	0.53	0.61	0.56
Others (C10)	0.34	0.51	0.41	0.53	0.47	0.50	0.69	0.66	0.67
Average	**0.39**	**0.52**	**0.44**	**0.54**	**0.49**	**0.51**	**0.71**	**0.68**	**0.70**
Accuracy	**0.59**			**0.63**			**0.81**		

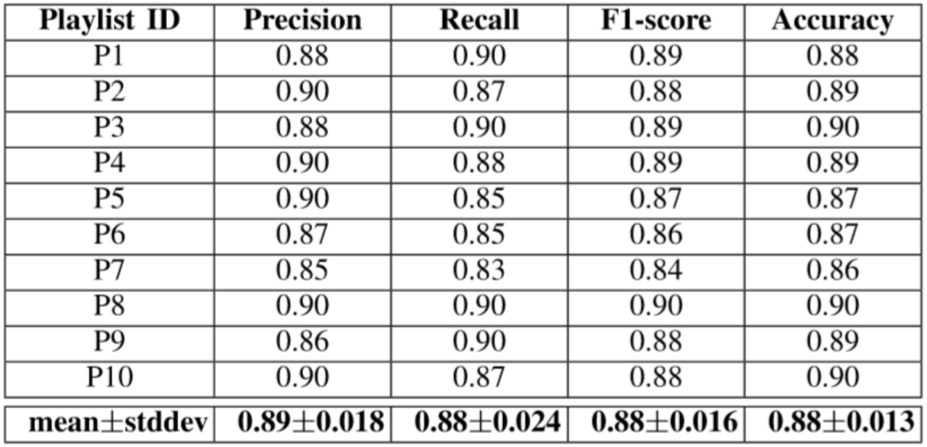


PERFORMANCE OF THREE INPUT STRATEGIES WITH LATE FUSION ARCHITECTURE

	Strategy-1			Strategy-2			Strategy-3		
Action Class	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Move cursor by keyboard (C1)	0.67	0.71	0.69	0.71	0.72	0.71	0.85	0.83	0.84
Move mouse over text region(C2)	0.74	0.60	0.66	0.72	0.61	0.66	0.87	0.85	0.86
Move mouse over non-text region (C3)	0.49	0.52	0.50	0.46	0.45	0.45	0.81	0.83	0.82
Enter text (C4)	0.49	0.40	0.44	0.53	0.51	0.52	0.81	0.84	0.82
Delete text (C5)	0.66	0.61	0.63	0.61	0.54	0.57	0.65	0.78	0.71
Trigger popups (C6)	0.50	0.46	0.48	0.57	0.38	0.45	0.75	0.83	0.79
Scroll text (C7)	0.45	0.40	0.42	0.47	0.41	0.44	0.75	0.61	0.67
Select text (C8)	0.65	0.47	0.54	0.65	0.46	0.54	0.67	0.80	0.78
Switch window (C9)	0.50	0.38	0.43	0.52	0.38	0.44	0.67	0.69	0.68
Others (C10)	0.41	0.63	0.49	0.43	0.58	0.49	0.74	0.68	0.70
Average	**0.45**	**0.62**	**0.52**	**0.47**	**0.58**	**0.51**	**0.75**	**0.70**	**0.73**
Accuracy	**0.60**			**0.62**			**0.82**		

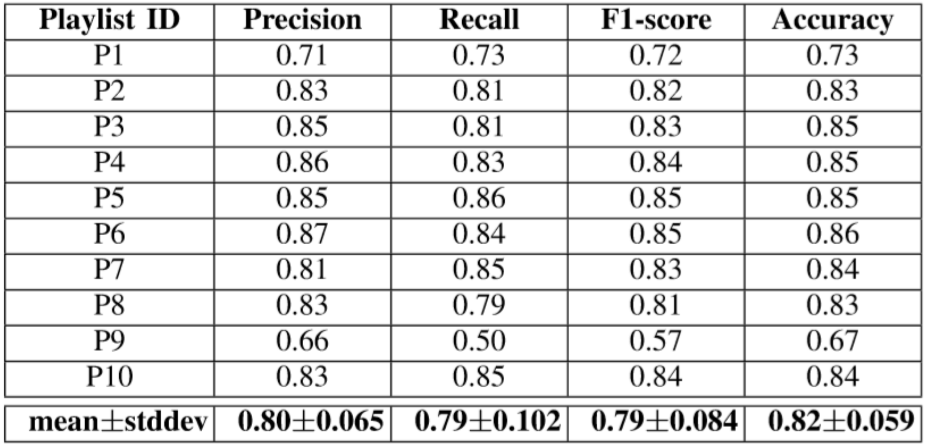
Evaluation

Intra Playlist Results



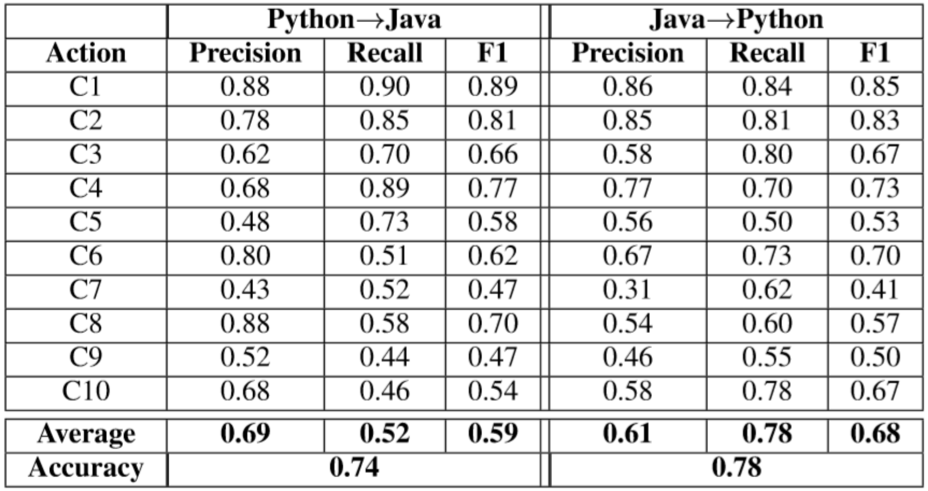
Playlist ID	Precision	Recall	F1-score	Accuracy
P1	0.88	0.90	0.89	0.88
P2	0.90	0.87	0.88	0.89
P3	0.88	0.90	0.89	0.90
P4	0.90	0.88	0.89	0.89
P5	0.90	0.85	0.87	0.87
P6	0.87	0.85	0.86	0.87
P7	0.85	0.83	0.84	0.86
P8	0.90	0.90	0.90	0.90
P9	0.86	0.90	0.88	0.89
P10	0.90	0.87	0.88	0.90
mean±stddev	**0.89±0.018**	**0.88±0.024**	**0.88±0.016**	**0.88±0.013**

Inter Playlist Results



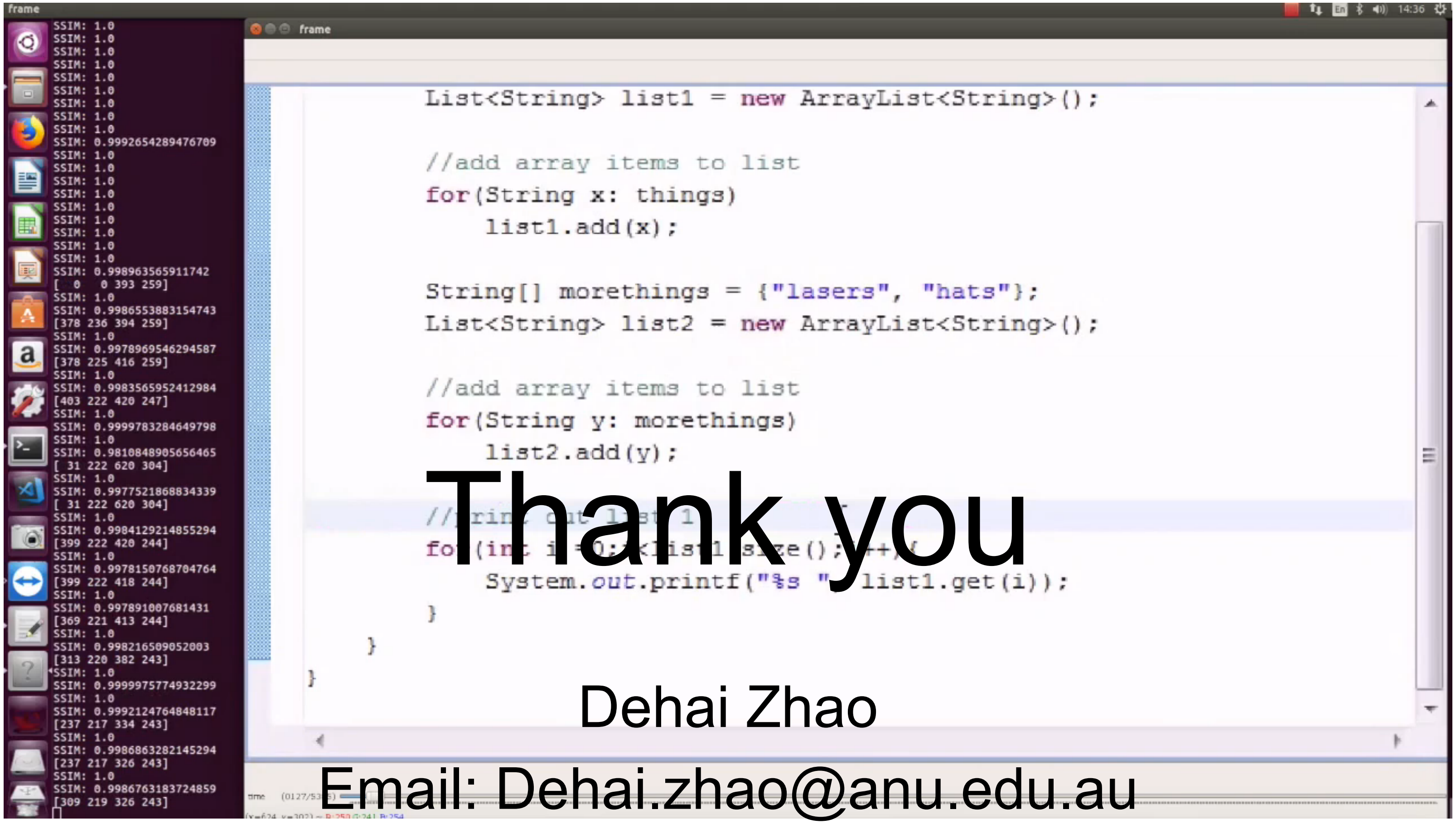
Playlist ID	Precision	Recall	F1-score	Accuracy
P1	0.71	0.73	0.72	0.73
P2	0.83	0.81	0.82	0.83
P3	0.85	0.81	0.83	0.85
P4	0.86	0.83	0.84	0.85
P5	0.85	0.86	0.85	0.85
P6	0.87	0.84	0.85	0.86
P7	0.81	0.85	0.83	0.84
P8	0.83	0.79	0.81	0.83
P9	0.66	0.50	0.57	0.67
P10	0.83	0.85	0.84	0.84
mean±stddev	**0.80±0.065**	**0.79±0.102**	**0.79±0.084**	**0.82±0.059**

Inter Programming Language Results



	Python→Java			**Java→Python**		
Action	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
C1	0.88	0.90	0.89	0.86	0.84	0.85
C2	0.78	0.85	0.81	0.85	0.81	0.83
C3	0.62	0.70	0.66	0.58	0.80	0.67
C4	0.68	0.89	0.77	0.77	0.70	0.73
C5	0.48	0.73	0.58	0.56	0.50	0.53
C6	0.80	0.51	0.62	0.67	0.73	0.70
C7	0.43	0.52	0.47	0.31	0.62	0.41
C8	0.88	0.58	0.70	0.54	0.60	0.57
C9	0.52	0.44	0.47	0.46	0.55	0.50
C10	0.68	0.46	0.54	0.58	0.78	0.67
Average	**0.69**	**0.52**	**0.59**	**0.61**	**0.78**	**0.68**
Accuracy	**0.74**			**0.78**		

Results



```
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 0.9992654289476709
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 1.0
SSIM: 0.998963565911742
[  0   0 393 259]
SSIM: 1.0
SSIM: 0.9986553883154743
[378 236 394 259]
SSIM: 1.0
SSIM: 0.9978969546294587
[378 225 416 259]
SSIM: 1.0
SSIM: 0.9983565952412984
[403 222 420 247]
SSIM: 1.0
SSIM: 0.9999783284649798
SSIM: 1.0
SSIM: 0.9810848905656465
[ 31 222 620 304]
SSIM: 1.0
SSIM: 0.9977521868834339
[ 31 222 620 304]
SSIM: 1.0
SSIM: 0.9984129214855294
[399 222 420 244]
SSIM: 1.0
SSIM: 0.9978150768704764
[399 222 418 244]
SSIM: 1.0
SSIM: 0.997891007681431
[369 221 413 244]
SSIM: 1.0
SSIM: 0.998216509052003
[313 220 382 243]
SSIM: 1.0
SSIM: 0.9999975774932299
SSIM: 1.0
SSIM: 0.9992124764848117
[237 217 334 243]
SSIM: 1.0
SSIM: 0.9986863282145294
[237 217 326 243]
SSIM: 1.0
SSIM: 0.9986763183724859
[309 219 326 243]
```

```java
        List<String> list1 = new ArrayList<String>();

        //add array items to list
        for(String x: things)
            list1.add(x);

        String[] morethings = {"lasers", "hats"};
        List<String> list2 = new ArrayList<String>();

        //add array items to list
        for(String y: morethings)
            list2.add(y);

        //print out list 1
        for(int i = 0;i<list1.size();i++){
            System.out.printf("%s ", list1.get(i));
        }
    }
}
```

Thank you

Dehai Zhao

Email: Dehai.zhao@anu.edu.au