Towards Making Flowchart Images Machine Interpretable

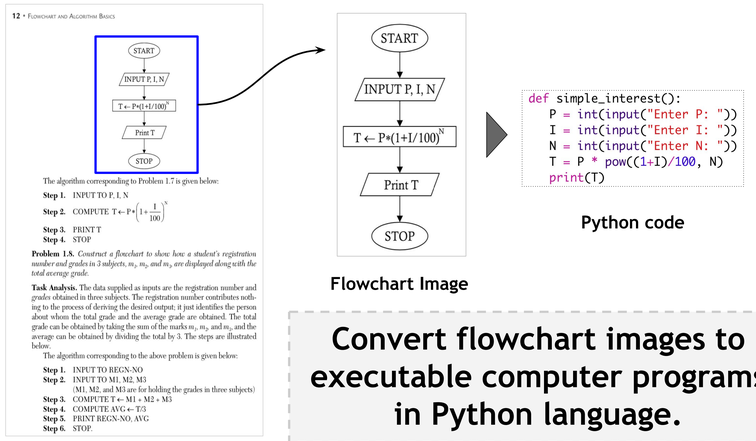
Shreya Shukla, Prajwal Gatti, Yogesh Kumar, Vikash Yadav, Anand Mishra
Vision, Language and Learning Group (VL2G)
IIT Jodhpur, India

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Data & Code available!

Let's transform visual logic into executable code with FloCo-T5!

GOAL



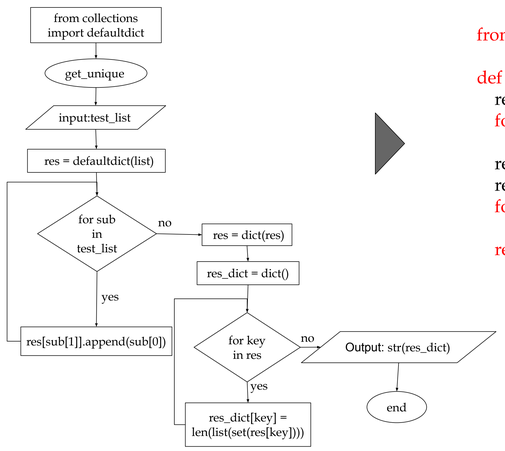
Flowchart Image

```
def simple_interest():
    P = int(input("Enter P: "))
    I = int(input("Enter I: "))
    N = int(input("Enter N: "))
    T = P * pow((1+I)/100, N)
    print(T)
```

Python code

Convert flowchart images to executable computer programs in Python language.

FloCo DATASET



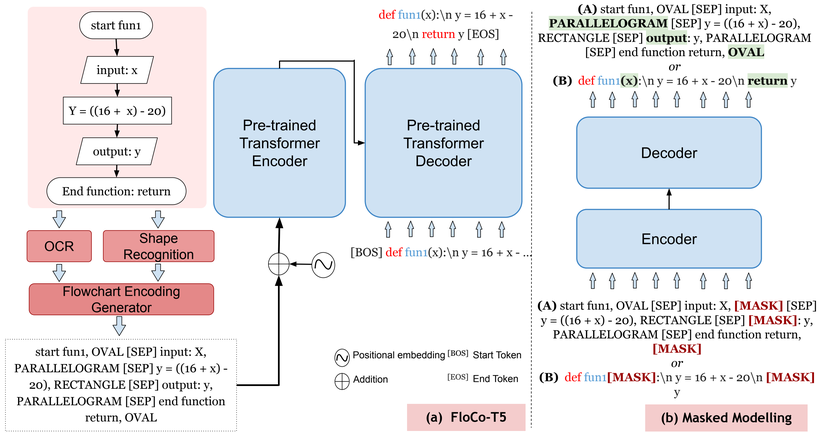

```
from collections import defaultdict

def get_unique(test_list):
    res = defaultdict(list)
    for sub in test_list:
        res[sub[1]].append(sub[0])
    res = dict(res)
    res_dict = dict()
    for key in res:
        res_dict[key] = len(list(set(res[key])))
    return str(res_dict)
```

A new dataset for studying our proposed task!
Contains 11K+ flowchart-code pairs.

Total Number of Samples	11884
Avg. length of the program (in tokens)	46
Avg. length of the program (in lines)	4.6
Train set size	10102
Test set size	1188
Validation set size	594

FloCo-T5



1. Flowchart image encoded using OCR and OpenCV shape detection results.
2. Masked Language Modelling objective used to train the pre-trained CodeT5 model on augmented python codes and flowchart image encodings.
3. Model fine-tuned with encoding-code samples.

Tuple Encodings	String Encodings	Modified String Encodings
[('start fun1', 'OVAL'), ('input: X', 'PARALLELOGRAM'), ('y = ((16 + x) - 20)', 'RECTANGLE'), ...]	{startfun1,OVAL}, {input: X,PARALLELOGRAM},{y = ((16 + x) - 20),RECTANGLE}, ...	start fun1, OVAL [SEP] input: X, PARALLELOGRAM [SEP] y = ((16 + x) - 20), RECTANGLE [SEP] ...

Types of encodings	BLEU	CodeBLEU	EM
Tuple	16.7	37.7	0.2
String	50.1	63.4	11.1
Modified String	**67.4**	**75.7**	**20.0**

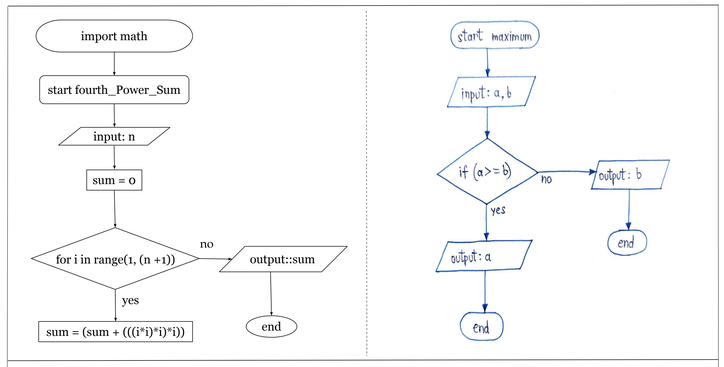
Experimented with 3 ways of encoding flowcharts

Original Code	Function augmented
import numpy as np def theta(self, s): s = np.where(s < -709, -709, s) return 1 / (1 + np.exp((-1) * s))	import numpy as np def he9GxMm5QgFn(self, s): s = np.where(s < -709, -709, s) return 1 / (1 + np.exp((-1) * s))
Variable augmented	**Function-variable augmented**
import numpy as np def theta(self, kO9): kO9 = np.where(kO9 < -709, -709, kO9) return 1 / (1 + np.exp((-1) * kO9))	import numpy as np def he9GxMm5QgFn(self, Lyv): Lyv = np.where(Lyv < -709, -709, Lyv) return 1 / (1 + np.exp((-1) * Lyv))

Logic-preserving code augmentations used during the pre-training of FloCo-T5

RESULTS

Method	BLEU	CodeBLEU	EM
Vanilla Transformer	10.3	26.8	0.0
BART	31.1	40.7	2.2
PLBART	55.7	63.7	19.7
CodeT5	63.8	71.8	17.8
FloCo-T5	**67.4**	**75.7**	**20.0**




```
def fourth_Power_Sum(n):
    sum = 0
    for i in range(1, n + 1):
        sum = sum + (i * i * i * i)
    return sum
```

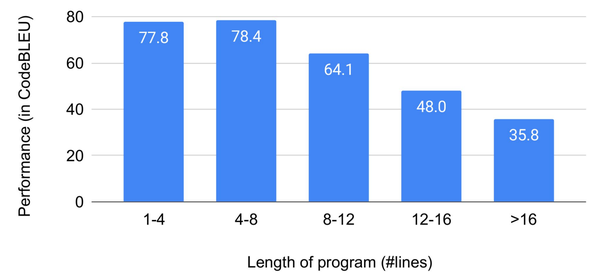
Ground Truth

```
def maximum(a, b):
    if a >= b:
        return a
    else:
        return b
```

```
def _fourthPowerSum(n):
    sum = 0
    for i in range(1, n+1):
        sum = sum + (i * i * i * i)
    return sum
```

FloCo-T5

```
def maximum(a, b):
    if a < b:
        return b
    return a
```



CONCLUSION

- A large-scale Testbed for Flowchart Image → Code Task
- Novel FloCo-T5 framework for code generation from flowchart images.
- Our model can be adopted to hand-drawn flowchart images as well.

SAN JOSE, CALIFORNIA, USA 2023

Acknowledgements: This work was partly supported by MeitY, Govt. of India (project number: S/MeitY/AM/20210114). Yogesh Kumar is supported by a UGC fellowship. We thank the ACM-W scholarship program sponsored by Google for supporting the conference travel.