Scan2Run
Reviving old listings in MAME emulator

PONSARD Christophe – FOSDEM 21 (online)
Context – NAM-IP Computer Museum

• Located in Namur/Belgium - 30’ from Brussels (worth a visit next FOSDEM when back at ULB)

• Missions:
  • Preservation: safeguarding digital heritage, focus on local pioneers
  • Acquisition of artefacts, enriching collections
  • Exhibition: for all, specific animation, permanent/temporary
  • Research: about machines, software, communities

• “Container design”, an historical parallel

www.nam-ip.be
Digital Preservation Constraints

• Machine in preservation mode:
  • Hard to use, not always multiple machines available, e.g. inDATA DAI rare Belgian computer
  ➔ benefit of using an emulator, e.g.

• Variety of (native) supports:
  • old floppies (8, 5 ¼ inch ...), magazines with listing,...
  ➔ focus here on listings

• Bottom line: SCAN listing to RUN in emulator
SCAN STEP – Listing Specificities

• listing “fonts” (e.g. dot matrix printer)
  • usually low res, not very good quality scan
  • aging issues

• listing not text!
  • statistical models based on corpus not useful
  • even degrading e.g. variable mixing letter/number ➔ numbers tend to become letters 1/I, 0/O
  • Initial numbers can be identified as a column
  • could train a “listing” corpus (e.g. keywords) possibly with some parsing rules

• reuse:
  • same printing style across pages of long listing
  • in multiple numbers of the same magazine,...
Scanning Listings – Selected Approach

• Open Source solution

• Avoid fully automated approach
  • User in loop but try to be involve it only when necessary
    unknow character, ambiguities, possible artefact...
  • Expected decreasing interactions “learning” phase ➔ “confirmation” phase

• Globally KISS approach – Keep It Simple Stupid (for now)
  • Avoid big black box not easy to control
    ➔ Tested but not very successful with Tesseract OCR
  • Simple understandable code base
    ➔ Character tracer ➔ recognition
    ➔ Take control over a “learning” loop
    ➔ JavaOCR (already used in a previous project)
  • Potential for improving later, e.g. feeding into neural network etc.
Logical Steps in original JavaOCR

- Character and sequencing: document → rows → characters

- Character is compared with training set and best match is selected (Mean Square Error used)

- NOT efficient: need to manually collect characters BEFORE starting scan

⇒ MORE DYNAMIC APPROACH
Scan2Run UI: build training set dynamically (if does not exist or reuse/improve previous)

• Learning phase (few know character) ➞ high match required (to avoid matching 8-S, 0/O etc)
• Then lower MSE threshold

```java
public double computeAskThreshold() {
    int s = trainingImages.keySet().size();
    if (s > 20) return askThreshold;
    return askThreshold - 0.025 * s;
}
```

• Also ask if second match is close to first match!

```java
boolean ambiguity = false;
if (bestCount > 1) diff = bestMSEs[1] - bestMSEs[0];
if (diff < 0.3) ambiguity = true;
```
Scan2Run (Improved) Design over JavaOCR

Interactive scanning application

- JFrame
- LearningGUI
  +main() +loadImage() +saveText() +loadTrainingSet() +saveTrainingSet() +clearTrainingSet() +start() +stop()

- Observer
  +selectionUpdated() +textUpdated() +userRequested() +scanFinished()

- ImageArea
- TextArea
- InteractionArea

Start/stop scanning
Update UI
User Feedback

Visitor

Forked JavaOCR

- DocumentScanner
  +scan() +addTrainingImages() +getTrainingSet()

- OCRScanner (in new thread)
  +start() +stopRequested()

- OCRScannerDemo
  +main()

- PixelImage
  +main()

- TrainingImage
  +main()
Demo Scan

```
1  MODE 0
2  REM M.DIERCKX
3  REM IMP INT
10  REM het maken van een doolhof
20  MODE 2:COLOR 0 8 8 0
80  FOR I=0 TO XMAX STEP 10:DRAW I,0 I,YMAX 21:NEXT
90  FOR I=0 TO YMAX STEP 10:DRAW 0,I XMAX,I 21:NEXT
130  EIND=RND(30)+60
135  X1=0:Y1=0
140  FOR I=1 TO EIND
150  IF INT(RND(2))=0 THEN FLAG=1:X=RND(XMAX):DRAW X1,Y1 X1 X1 0:
160  IF FLAG=0 THEN Y=RND(YMAX):DRAW X1,Y1 X1,Y1 0:Y1=Y
165  FLAG=0
170  NEXT I
175  DRAW X1,Y1 X1,YMAX 0
176  DRAW X1,YMAX XMAX,YMAX 0
```
Performance: % of requests, % of errors?

- Depends on scan quality!
  - Use good scan quality (>= 300 DPI)

- Some internal preprocessing but better to:
  - Remove non listing part, artefacts
  - Enhance contrast

- Typical requests:
  - Medium quality 10% requests on 1 page listing with no learning set
  - High quality, more pages: can go down to 3% ➔ easier than retyping

- Error rate:
  - Not assessed exactly so far: partial visual inspection
  - Can be detected when injecting (syntax error) or running (not expected behaviour)
  - Very good listing ➔ no error but usually some systematic (ambiguity) ➔ correct or adapt parameter

- Know limitations with workaround or needing enhancement
  - Character tracer can merge some character ➔ possible to encode more than 1 char at request
  - Small punctuation characters have high MSE probably due to small size, need to “normalize”? ➔
  - Row problem ➔ need to adjust internal parameter (not available at UI) or to edit scan (more heavy)
RUN STEP - Scripting MAME to Inject Code

• Use MAME in console mode with LUA scripting
  
mame -console -autoboot_script <import_functions>.lua

• LUA scripting functionalities
  • usual functions, e.g. for file IO to guest system
  • list configuration
  • pause/unpause
  • I/O: access memory, post key events, write on screen, K7,...
  • various listeners: frame, frameDone, Sound, Periodic
  • ...

• Simplest design: send lines
  • assume some buffering if emulator limits input
  • In two lines:
    f=io.open(path)
    for line in f:lines() do emu.keypost(line.."\n") end
Injecting into MAME - DEMO

MOD=8
basic_s=""
basic_l=0;

function readAll()
    local f=io.open("C:\DEV\NAM-IP-WEB\DAI\CPC_BOMB.BAS")
    basic_s = f:read("*all")
    basic_l = string.len(basic_s)
end

function basic_load()
    readAll();
end

i=0;
j=0;
function basic_post()
    if ((i%MOD==0) and (j<=basic_l)) then
        emu.keypost(c);
        io.write(c);
        j=j+1;
    end
    i=i+1;
end

emu.pause();
readAll();
emu.unpause();
emu.register_frame(basic_post)

Will write to MAME every MOD frame
Can be tuned to “match” emulator ability to capture character

Fix developed for some problems with DAI (but did not work actually, see other workaround)
Demo INJECTION and RUN
Some recovered programs on DAI (from DAINAMIC magazine)

- Math 3D plotter
- Radar simulator (with sound)
- MAZE Game
Conclusion / Next Steps

• Try it from here – beware this is still work in progress
  https://github.com/NAMIP-Computer-Museum/scan2run

• Quite “BASIC” approach but gets the work done for such BASIC programs ;-) 

• Architecture quite easy to master
  ➔ possible to extend with more powerful learning capabilities!

• Feedback / ideas / contributions welcome!