Solid Framework
NSE Quality Improvements

Reconstruction and API Improvements in Build 10278

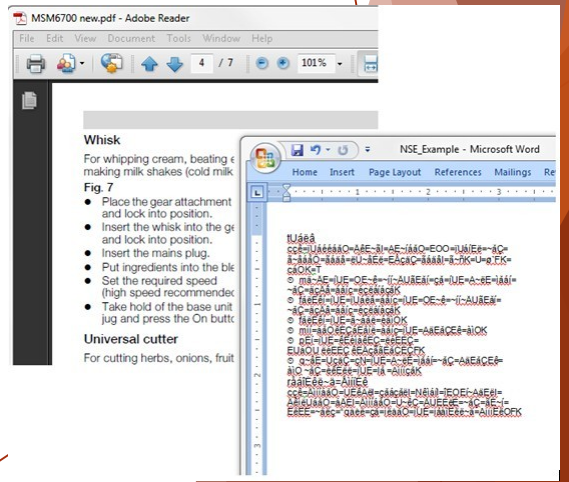
Release Date – 4th June 2020

What is NSE? – Non Standard Encoding

What does that mean?
There is a really good explanation at
http://blog.soliddocuments.com/2010/06/what-is-non-standard-encoding.html

In summary, there is no requirement that the characters in the fonts used in a PDF have to look like the characters that they represent.

Why does that matter?
At its simplest, if you try to copy from such a PDF the output is different, often VERY different, and may be of zero value.



Solid Framework and NSE

- Solid Framework has supported NSE detection for many years.
- We aim to make NSE detection automatic, so that “It just works” without the user needing to do anything.
- It is available as part of the standard Solid Framework distribution – it does not require an OCR license.

This Presentation describes a few of the improvements in Solid Framework NSE handling that were introduced in 10.0.10278

General Improvements

Solid Framework was already good at detecting problems with English Language text.

Further improvements have been made with English.

In addition, significant improvements have been made with other languages.

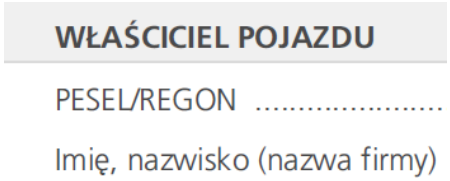
Better NSE Detection

PDF	10158	10278
DISTRIBUCIÓN	D[illegible]	DISTRIBUCIÓN
	No useful text recovered	Text is correct

929bf7e0aa9ab91f3554ee117c7942cc969e4c85

Better Detection of Polish Characters

PDF



10158

Characters incorrect

10278

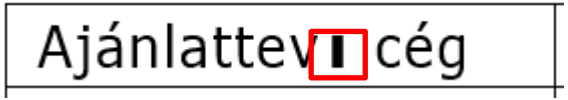
Characters correctly recovered

9e5c43ad32d76c2692390a5e15ed4cc3a6f68676

Better Detection of Hungarian Characters

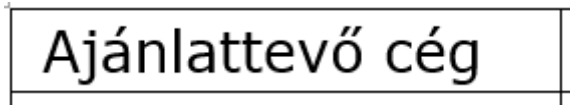
PDF

10158



Character incorrect

10278



Character correctly detected

d7726fb6446ac9789e772a733cc92ce225166661

Better Detection of Hungarian Characters

PDF

Tárgy: előzetes

10158

Character incorrect

10278

Tárgy: előzetes

Character correctly detected

e71591fb4eeb0ece10ff502a1f9c57277168ddbc

All Caps Text

Some fonts contain upper and lower case characters that look identical

This means that they could be encoded in either case and look the same.

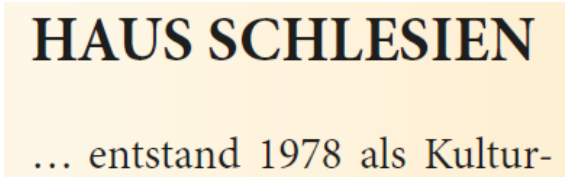
If the font does not exist on the user's machine then it will be substituted by one that does. If this is *not* an all caps font (they are rare) then this could result in mixed case output.

The user, however, wants the reconstructed document to look like the PDF – not its encoding.

To avoid this Solid Framework detects that the "lower case" characters actually should be represented by uppercase characters in the substitution font.

Improved Reconstruction of All Caps Text

PDF



10158

Many of the letters are reconstructed as lower case

10278

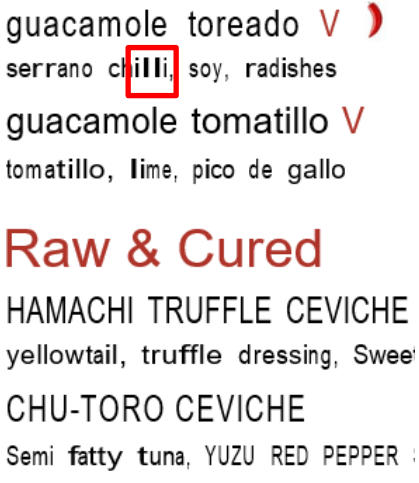
Characters are reconstructed as all caps

d4c44af413fcf7a3f7f1a63a8a736d17e7b6d2a7

Improved Reconstruction of All Caps Text

PDF

10158



Many letters reconstructed as lower case, resulting in the “l” characters looking too wide.

10278

Characters are reconstructed as all caps

41c7fbcd974aa8269208bbe10fff6e99af09f6ed

Small Caps Text

Some fonts actually contains dedicated characters that are small capitals.

Word, however, uses upper case characters that have been scaled down to give the effect.

In either case the user wants the reconstructed document to look like the PDF.

This can be difficult since some characters look the same in both upper and lower case – and it is only their size that indicates what they should be considered to be.

All of the characters except the “T” are lower case “small capitals”

Small Caps Text – Example 1

PDF

The letters are encoded as a mixture of upper and lower case characters

10158

Many of the letters are reconstructed as lower case (since that is how they were encoded in the PDF)

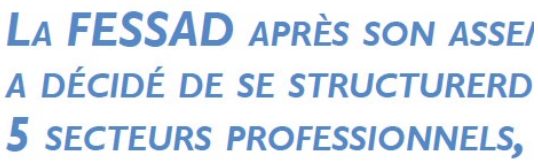
10278

Characters are reconstructed as small caps

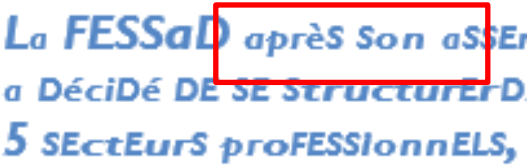
5fba67f4285b2b4e323dec133ca24c4798f734a3

Small Caps Text – Example 2

PDF



10158



Many of the letters are reconstructed as lower case

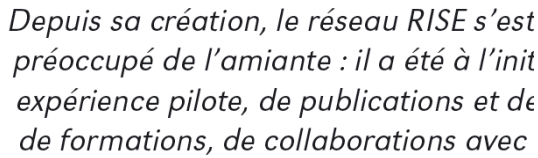
10278

Characters are correctly reconstructed as small caps

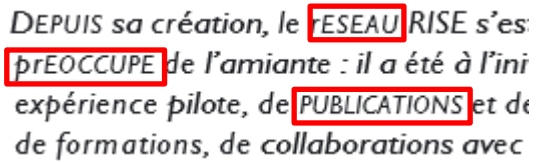
db9557ecef2580a9227e7114717ccd5d21a3b23b

Small Caps No Longer Wrongly Detected

PDF

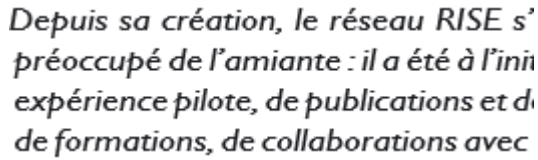


10158



Many of the letters are incorrectly reconstructed as small caps

10278



Characters are correctly reconstructed as lower case

454bc9ea2acc3a3409fecb2ce1ccab78d1e74fb3

Small Caps No Longer Wrongly Detected

PDF

Around 90% of students in OECD
places where students can learn abou

In most countries, science-related
to better student performance, a

10158

AROUND 90% of STUDENTS in OECD
places where STUDENTS can learn ABO

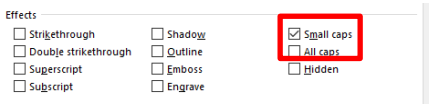
In most COUNTRIES, science-related
to better STUDENT performance, a

10278

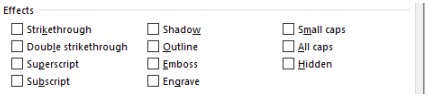
Around 90% of students in OECD
places where students can learn abou

In most countries, science-related
to better student performance, a

Several words are incorrectly detected as Small Caps



Characters are correctly reconstructed as lower case



0b5a778ad7be4c0370e814ab64f863270eed70e9

Small Caps No Longer Wrongly Detected

PDF

Fühler für technische

10158

FÜHLER FÜR technische

Small Caps incorrectly detected for some characters

10278

Fühler für technische

Characters are correctly reconstructed as lower case

0d519bd35150ecb1ee665590295df179d6005324

Ligatures

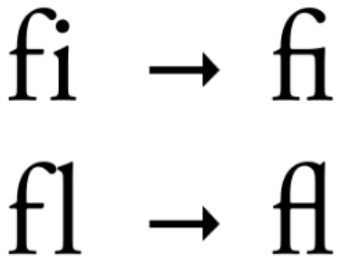
What is a ligature?
A group of letters, displayed as a single character for aesthetic reasons (usually to control the spacing between characters)

Standard Ligatures are “ff”, “fi”, “fl”, “ffl” and “ffi”.

- These ligatures have Unicode codepoints and exist in most fonts

Other ligatures – can be almost anything. Common ones are “ft” and “tt”.

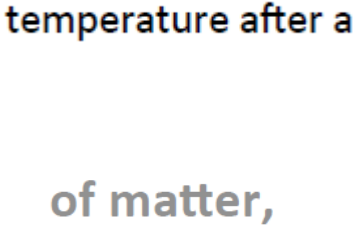
- These DO NOT have Unicode codepoints



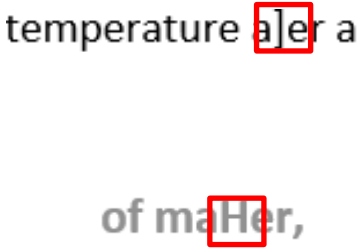
See https://en.wikipedia.org/wiki/Orthographic_ligature for more information

Better Detection of Ligatures

PDF

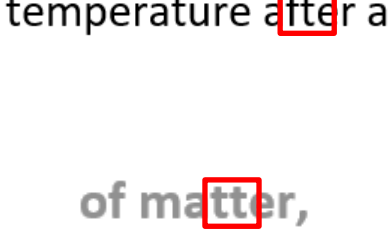


10158



“ft” and “tt” ligatures are incorrect

10278

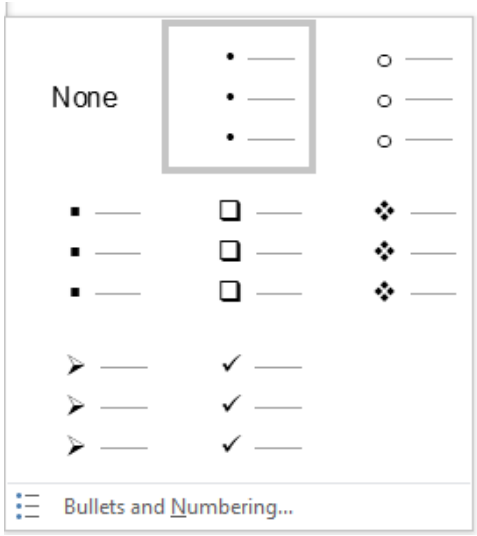


Ligatures correctly detected

738e5f97d6b6ef97a5fc33c541917b8aa2f2d311

Bullets

- “Bullets” are used to indicate list items.
- The actual symbol can be almost anything.
- This makes detection of bullets a mixture of correct character identification, and correct interpretation.



Bullet symbols available in PowerPoint

Improved Bullet detection

PDF

- Languages:
  - German (native speaker)
  - English (fluent written and spoken)
  - Dutch (advanced)

10158

- Languages:
  - German (native speaker)
  - English (fluent written and spoken)
  - Dutch (advanced)

Bullet character is incorrect

10278

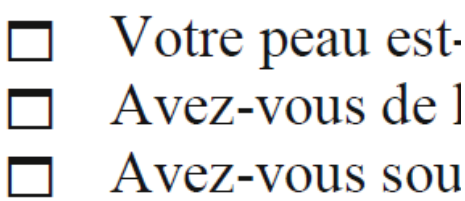
- Languages:
  - German (native speaker)
  - English (fluent written and spoken)
  - Dutch (advanced)

Bullets detected correctly

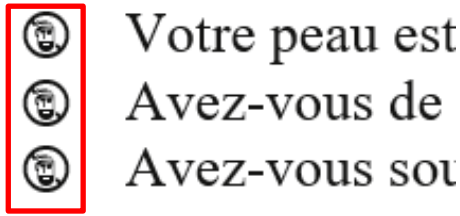
b679bd9eaa72a8edadb9ab47d2e27c107e27a6b2

Improved Bullet detection

PDF

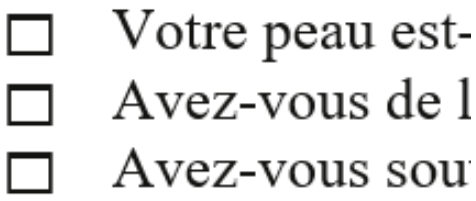


10158



Wrong symbol used

10226



Correct symbol used

73289189e76e0588285c97f9f7703400a7956864

Improved Bullet Detection

PDF

10158

Bullet incorrect

10278

Correct detection of bullet

c93aa267a7135ee1eb821ae03a7bd036f57336b1

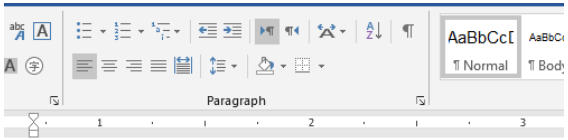
Improved Bullet Detection

PDF

■ PLIKI

Przyjmujemy materiały

10158

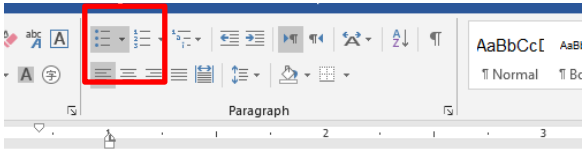


④ PLIKI

Przyjmujemy materiały

Bullet incorrect

10278



■ PLIKI

Przyjmujemy materiały

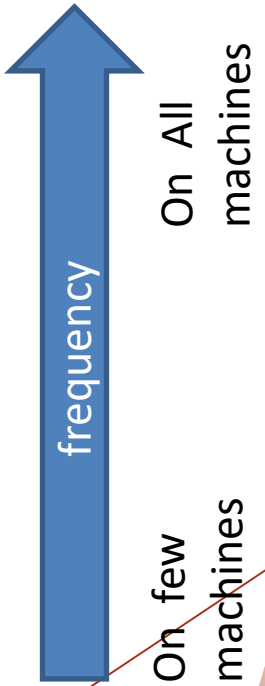
Correct detection of bullet helps to correctly identify the presence of a list

565caedce9821c842846a8ab74c6df7671dbc909

Non Alphanumeric Characters

Various types of non alphabetic characters exist:

- Common Symbols - exist in common fonts on end-user machines like Symbol, Wingdings and Webdings
- Uncommon Symbols - fonts like ZapfDingbats which exist on a fraction of end-user machines (Mac) and are better represented using other Unicode fonts (like Arial Unicode MS)
- Barcodes - special fonts that are unlikely to exist on end-user machines and content that is unlikely to need editing - better to preserve exact appearance as image
- Icons - specialized pure icon fonts used by designers but highly unlikely to be on end-user machines - also content that will more likely be replaced than edited so better preserved by rendering as a vector graphic

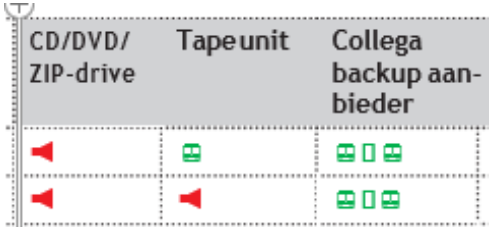


Improved Symbol Detection of Webdings Font

PDF

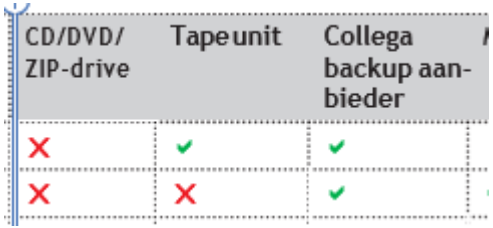
CD/DVD/ ZIP-drive	Tape unit	Collega backup aan- bieder
✗	✓	✓
✗	✗	✓

10158



Characters are incorrect

10278

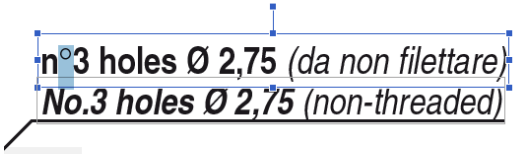


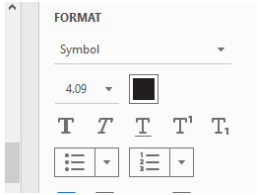
Correct characters are detected

5bc295c560e5c0c2937570824f3522569b58d7f7

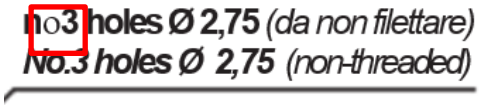
Better Detection of Symbols

PDF



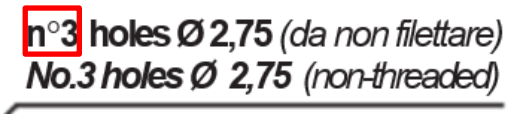


10158



Character incorrect. It should be a degree symbol, not an “o”

10278

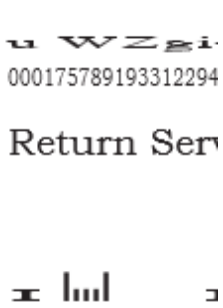


Character correctly detected and located

Better Handling of Barcodes

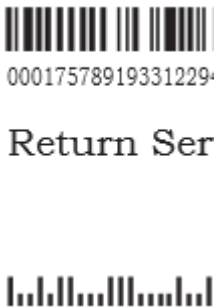
PDF

10158



Barcode font is considered to be text making it useless

10278



Barcode is rendered as an image, resulting in something that looks correct

995db644005a52ec877d93ab4e653897964dd8f6

Better Detection of Unusual characters

PDF

The clock is a character from Wingdings font encoded as “periodcentered”

10158

The clock is lost, and replaced by the “periodcentered” character

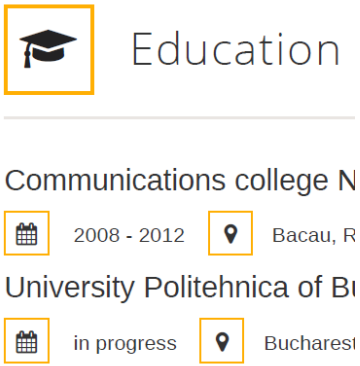
10278

Clock is correctly reconstructed

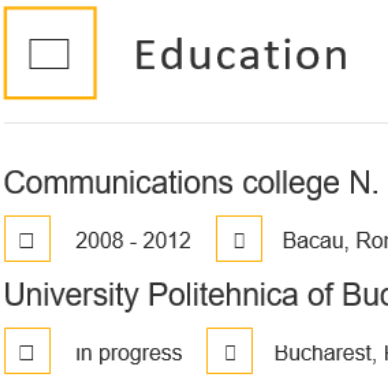
de850da88b0b1aa7f5ae3b3ee5ce519b0a684226.pdf

Better Detection of Unusual characters

PDF

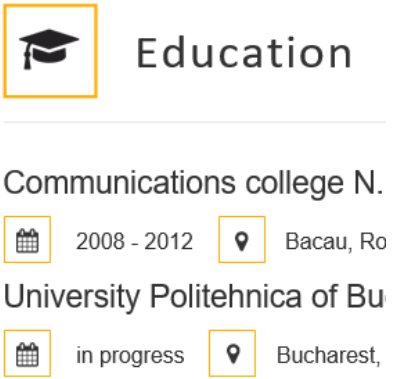


10158



Characters incorrect

10278

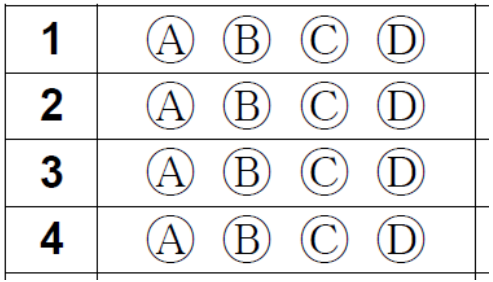


Correct characters are detected

5c775d579ff5ee99aeacb1eb0f28bddf83794d53.pdf

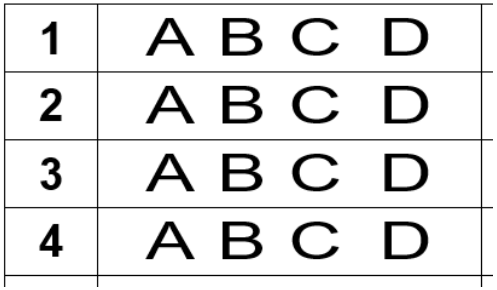
Better Detection of Unusual characters

PDF



1	Ⓐ	Ⓑ	Ⓒ	Ⓓ
2	Ⓐ	Ⓑ	Ⓒ	Ⓓ
3	Ⓐ	Ⓑ	Ⓒ	Ⓓ
4	Ⓐ	Ⓑ	Ⓒ	Ⓓ

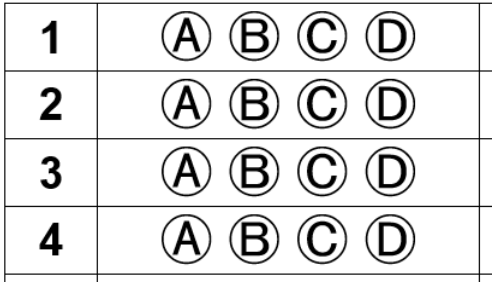
10158



1	A	B	C	D
2	A	B	C	D
3	A	B	C	D
4	A	B	C	D

The letters are recovered, but it's not really right

10278



1	Ⓐ	Ⓑ	Ⓒ	Ⓓ
2	Ⓐ	Ⓑ	Ⓒ	Ⓓ
3	Ⓐ	Ⓑ	Ⓒ	Ⓓ
4	Ⓐ	Ⓑ	Ⓒ	Ⓓ

Reconstruction is correct

f4f0cd68c35a2f90361f3cb249dceabf87286bd1.pdf

New Option when Converting to Text – PreserveRareUnicode (default false)

Unicode supports a number of characters that are valid, but may be confusing.

- Ligatures
- Some bullets
- Small caps

These are not supported by all fonts.

The option PreserveRareUnicode allows you to decide whether or not more common look-alike characters should be used instead when these poorly supported Unicode codepoints are detected.

This will be described further in a separate presentation.

Preserve Rare Unicode

PDF

Default

```
MIKE GULDIN AND
ROLLIN’ & TUMBLIN’
```

Reconstruction uses characters found in most fonts

PreserveRareUnicode=true

```
MIKE GULDIN AND
ROLLIN’ & TUMBLIN’
```

Reconstruction uses Unicode Small Caps characters

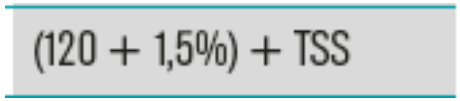
29735872b7c08eeb936908455898c790f4834690

Other Improvements

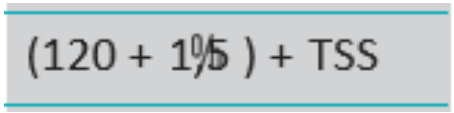
Better NSE detection has resulted in improved layout detection

Better NSE Detection Leads to Better Layout

PDF

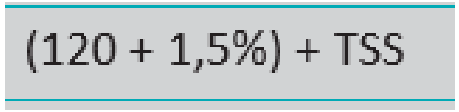


10158



The text is incorrectly laid out

10278

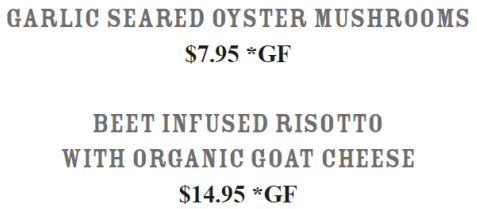


The conversion is correct

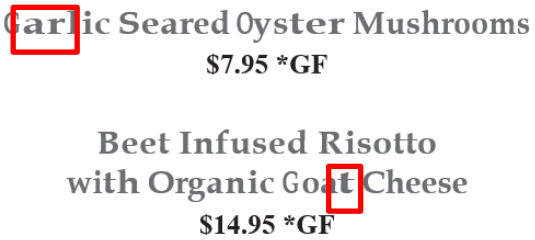
0425b706c8b23820615245fd4514f9ab4643cc8f

Better All Caps Detection Leads to Better Layout

PDF

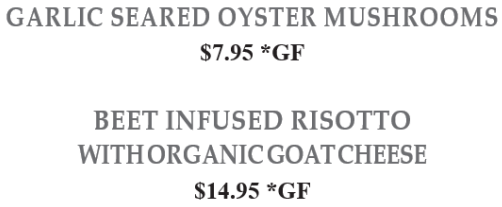


10158



Many characters are detected as lower case. Some have to be “stretched” to make them the same size as the characters in the PDF.

10278



All caps are correctly detected, resulting in a much better looking reconstruction.

45457a299fbc31e8536dd61292446692335e9e91