

## Trabalho de Formatura Supervisionado

# Aplicação de análise morfológica para segmentação de páginas em imagens de documentos

Ricardo de Cillo

Supervisora: Nina S. T. Hirata

Departamento de Ciência da Computação  
Instituto de Matemática e Estatística, IME-USP

**Resumo:** Neste texto apresentamos nosso estudo sobre a aplicação de operadores morfológicos à segmentação de páginas de documentos, etapa importante na análise de documentos que busca extrair informações sobre a sua estrutura: regiões com títulos, legendas, figuras e blocos de texto. A qualidade da solução obtida será medida e comparada, segundo os mesmo critérios aplicados à resultados considerados estado da arte por pesquisadores da área.

São Paulo, 10 de fevereiro de 2013

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Fundamentos</b>	<b>2</b>
2.1	Imagens digitais . . . . .	2
2.2	Operadores de imagens . . . . .	2
2.2.1	Binarização de imagens . . . . .	2
2.3	Classificação de objetos . . . . .	2
2.4	Segmentação de imagens . . . . .	3
2.5	Classificação dos componentes . . . . .	3
2.6	Avaliação da segmentação . . . . .	4
<b>3</b>	<b>Aprendizado de operadores morfológicos</b>	<b>4</b>
3.1	Operadores morfológicos binários . . . . .	4
3.2	Treinamento . . . . .	4
3.2.1	Coleta . . . . .	5
3.2.2	Decisão . . . . .	5
3.2.3	Minimização . . . . .	5
<b>4</b>	<b>Metodologia</b>	<b>5</b>
4.1	Preparação das imagens de treinamento . . . . .	5
4.2	Construção dos operadores . . . . .	6
4.3	Aplicação dos operadores . . . . .	6
4.4	Consensualização . . . . .	6
<b>5</b>	<b>Experimentos</b>	<b>6</b>
5.1	Base de dados . . . . .	7
5.2	Mistura de publicações no conjunto de treinamento . . . . .	7
5.3	Tipos de regiões . . . . .	7
5.4	Quantidade de imagens de treinamento . . . . .	8
5.5	Formatos de janelas . . . . .	8
5.6	Tamanho da janela de consensualização . . . . .	8
5.7	Aplicação . . . . .	9
<b>6</b>	<b>Resultados</b>	<b>9</b>
<b>7</b>	<b>Conclusão</b>	<b>15</b>
<b>8</b>	<b>Apêndice</b>	<b>15</b>
8.1	TRIOS . . . . .	15
8.2	Algoritmo de Otsu para binarização . . . . .	23

# 1 Introdução

Processamento e análise de documentos é uma importante subárea da área de reconhecimento de padrões cujo principal objetivo é a interpretação de um documento, ou seja, o entendimento da sua estrutura bem como o reconhecimento de cada um dos componentes estruturais.

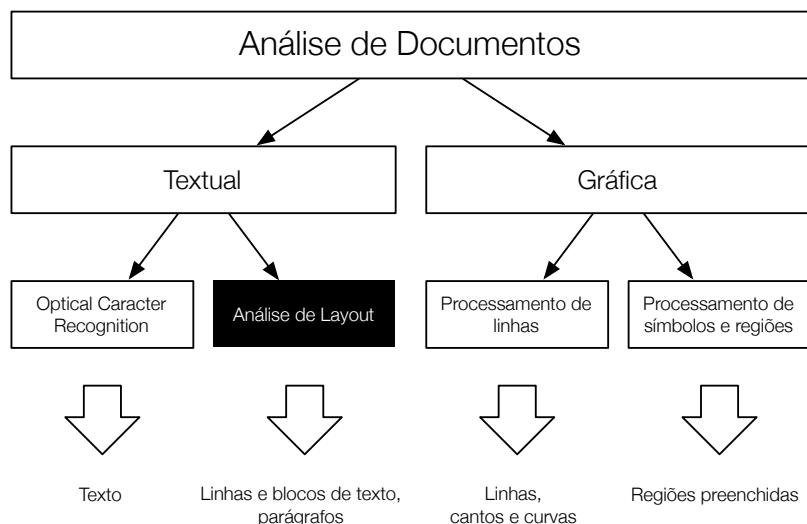


Figura 1: Contextualização do tema do trabalho entre as áreas da análise de documentos. Adaptado de [1].

Segmentação de página refere-se à tarefa de separar e rotular os diferentes componentes que fazem parte da estrutura das páginas de um documento, tais como: blocos de texto, gráficos, figuras, títulos, legendas, separadores, tabelas, fórmulas matemáticas e regiões com ruído.

Em geral, a segmentação de página é um dos primeiros passos no processo de entendimento de um documento. Uma vez identificados os blocos estruturais, processamentos específicos para cada tipo de bloco podem ser aplicados. Por exemplo, no caso de blocos de textos é conveniente fazer o reconhecimento de texto para que o mesmo possa ser armazenado em formato texto (e não imagem). Por outro lado, no caso de imagens, pode ser interessante armazená-las em alta resolução para manter a qualidade. Documentos digitalizados podem ser processados eficientemente em processos que envolvem armazenamento, edição, transmissão, ou busca, por exemplo.

Devido à grande quantidade de documentos, é interessante que o seu processamento seja realizado de forma automatizada ou pelo menos semi-automatizada. Para tal, diversas soluções computacionais vêm sendo propostas para o problema ao longo dos anos desde o surgimento desse campo de pesquisa. Automatizar esta tarefa reduz custos, aumenta a velocidade e capacidade de processamento de documentos além de possivelmente reduzir a taxa de erro humano na classificação de uma região.

Neste trabalho exploraremos a aplicabilidade de operadores morfológicos automaticamente gerados ao problema de segmentação de páginas.

## completar a descrição da estrutura do texto

Este texto está organizado da seguinte forma. Na seção 2, apresentamos as definições e conceitos básicos que serão importantes para a leitura deste texto.

## 2 Fundamentos

### 2.1 Imagens digitais

Uma imagem digital monocromática pode ser definida como uma função  $f : \mathbb{E} \subset \mathbb{Z}^2 \rightarrow \mathbb{K} = \{0, 1, \dots, k-1\}$ , na qual  $k$  representa o número de tons de cinza. Tipicamente adota-se  $k = 256$ , ou seja, 8-bits de cor. Quando  $k = 1$  as imagens são denominadas **binárias**; quando  $k > 1$  as imagens são denominadas **tons de cinza**. Na prática, o domínio  $\mathbb{E}$  é um retângulo finito de dimensões  $m \times n$  (uma matriz de  $m$  linhas e  $n$  colunas).

Uma imagem RGB (colorida) é uma função  $f : \mathbb{E} \rightarrow \mathbb{K}^3$ , onde cada componente  $\mathbb{K}$  representa a intensidade das cores vermelho, verde e azul, respectivamente.

### 2.2 Operadores de imagens

Um operador de imagens é uma função que mapeia imagens em imagens. Denotando  $\mathbb{E} = \mathbb{Z}^2$ ,  $K = \{0, 1, \dots, k-1\}$  e todas as imagens definidas em  $\mathbb{E}$  por  $K^{\mathbb{E}}$ , podemos representar um operador de imagens como  $\Psi : K^{\mathbb{E}} \rightarrow K^{\mathbb{E}}$ .

#### 2.2.1 Binarização de imagens

A classe de operadores estudada restringe-se ao domínio das imagens binárias. Porém as imagens obtidas através de digitalização usualmente são coloridas (RGB de 24-bits). O processo de binarização é realizado por um operador que mapeia uma imagem colorida ou monocromática em uma imagem binária.

Neste trabalho, primeiramente transformamos as imagens coloridas para níveis de cinza e posteriormente aplicamos a binarização.

Existem muitos algoritmos que realizam esta tarefa. Uma revisão extensa dos mais conhecidos métodos de binarização é apresentada em [2]. Todos eles se aplicam a imagens em níveis de cinza, portanto inicialmente transformaremos a imagem colorida  $f$  em níveis de cinza  $g$ :

$$f(x) \in \mathbb{K}^3 \rightsquigarrow g(x) \in \mathbb{K} \rightsquigarrow b(x) \in \{0, 1\} \quad (1)$$

### 2.3 Classificação de objetos

Na área de reconhecimento de padrões e aprendizado computacional estudam-se métodos e técnicas para classificação de dados em geral. Os dados (padrões) a serem classificados correspondem, em geral, à representação digital de algum objeto concreto ou abstrato. O objetivo da classificação é atribuir um rótulo de classe a cada padrão observado.

Dependendo do problema, os rótulos de classe podem ser conhecidos ou não. Por exemplo, se desejamos fazer o reconhecimento de caracteres, os padrões são a imagem dos caracteres e os rótulos de classe são as identificações dos possíveis caracteres. Por outro lado, em problemas como na classificação de perfil de consumidores, pode não haver um conjunto de perfis pré-estabelecidos e o objetivo seria então identificar a possível existência de perfis. O primeiro é conhecido como problema de classificação supervisionada e o segundo como classificação não-supervisionada.

No caso da classificação supervisionada, supõe-se que os padrões são elementos de um espaço  $X$  e que o conjunto de rótulo de classe é dado por  $Y = \{y_1, y_2, \dots, y_c\}$ . Assim, um classificador pode ser expresso por uma função  $f : X \rightarrow Y$ .

Frequentemente  $X$  é um subespaço de  $\mathbb{R}^d$ . Assim, um padrão é representado por uma  $d$ -upla  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ .

Tabela 1: Exemplo de segmentação de imagem: separando frente e fundo.



## 2.4 Segmentação de imagens

A segmentação de imagens é um processamento comum a praticamente todos os processos que envolvem análise de imagens. Segmentar uma imagem corresponde a particionar o seu domínio, de forma que cada região resultante corresponda (do ponto de vista semântico) a uma componente de interesse na análise em questão. Este problema pode ser modelado como uma classificação de objetos, onde o conjunto de pixels de uma imagem são os objetos em  $X$  e as componentes em  $Y$  são regiões de interesse, como ilustrado na tabela 1.

## 2.5 Classificação dos componentes

O problema de segmentação de imagens de documentos pode ser modelado como um problema de classificação de objetos, onde cada pixel da imagem é rotulado através de uma função classificadora

$$\Psi: \mathbb{E} \rightarrow \mathbb{Y} \quad (2)$$

sendo  $\mathbb{Y}$  um conjunto composto pelas regiões de interesse:

- blocos de texto: região com parágrafos
- gráficos
- figuras
- títulos
- legendas
- separadores
- tabelas
- fórmulas matemáticas
- regiões com ruído

## 2.6 Avaliação da segmentação

Dizer aqui como se avalia uma segmentação. Poderia ser a nível de pixel, ou a nível de regiões ou componentes. Pixel é fácil, mas tem os potenciais problemas. No caso de região é preciso definir como comparar regiões — caso dos polígonos isotéticos, etc etc

Os dois principais métodos utilizados para avaliar a qualidade das soluções para este problema são a comparação da classificação dos pixels individualmente ou a classificação de regiões. Optamos pela comparação entre pixels pois nosso método produz como resultado a classificação de pixels e não a delimitação de regiões. Seria necessário agrupar os pixels em regiões para que pudessemos realizar outros tipo de avaliação, o que foge ao escopo deste trabalho.

Uma vantagem na comparação entre pixels e não regiões é o fato de que evitamos restrições arbitrárias na forma das regiões.

## 3 Aprendizado de operadores morfológicos

Construir operadores morfológicos que resolvam problemas complexos como o de segmentar uma página de documento, pode ser uma tarefa que demande muito tempo, experiência e conhecimento específico do assunto. Como estas imagens possuem características distintas dependendo da publicação, é possível que apenas um operador não consiga ser aplicado a todas as imagens. Ou seja, construir operadores com facilidade é um fator sensível para a escolha desta abordagem.

Nesta seção apresentamos um método para projetar operadores morfológicos de forma automática utilizando técnicas de aprendizado computacional. Primeiramente definiremos o conceito de operadores morfológicos.

### 3.1 Operadores morfológicos binários

Informalmente, operadores morfológicos são transformações entre imagens binárias que podem ser descritas através de propriedades geométricas e topológicas. Um exemplo amplamente conhecido é o da dilatação, cuja aplicação está exemplificada na figura [ref](#).

[figura com dilatação e erosão](#)

Formalmente define-se um operador morfológico binário como uma transformação localmente definida e invariante por translação entre reticulados completos.

Um dos principais resultados da morfologia matemática revela que qualquer operador binário pode ser decomposto em uma combinação de operadores mais simples, dilatação (3) e erosão (4), chamados de elementares.

$$\delta_B(X) = \{x \in E : B_x \cap X \neq \emptyset\} \quad (3)$$

$$\varepsilon_B(X) = \{x \in E : B_x \subseteq X\} \quad (4)$$

O conjunto  $B$  é denominado elemento estruturante.

### 3.2 Treinamento

A fim de projetar operadores morfológicos complexos de forma automática, utilizaremos o método proposto em [ref](#). Nele, um conjunto de pares de imagens de treinamento é transformado em um operador. Estes pares são compostos da imagem **original**, denotada por  $X$  e sua **derivada**  $Y$ , que exemplifica a transformação desejada. ([figura ref](#)).

O processo consiste de 3 etapas (diagrama [ref](#)): coleta, decisão e minimização. Ao fim, teremos uma função booleana que aproxima é uma aproximação do operador desejado. A seguir detalhamos cada uma das etapas.

### 3.2.1 Coleta

Seja  $C(X_W) = \{X \cap W + z : z \in E\}$  o conjunto de todas as configurações observadas ao se transladar a janela  $W$  sobre a imagem  $X$ . Seja  $Y_z$  o valor 0 ou 1 encontrado na imagem  $Y$  na posição  $z$ . Construímos uma tabela com três colunas:  $C(X_W)$ , frequência observada  $Y_z = 0$  e frequência  $Y_z = 1$ . (tabela exemplo [ref](#))

### 3.2.2 Decisão

As configurações observadas na etapa anterior possuirão valores relacionados em  $Y$  iguais a 0, 1 ou ambos. No caso de tanto 1 como 0 terem sido observados, escolheremos como valor para a função final a com maior número de ocorrências. No caso de alguma configuração não ter sido observada ou de o número de ocorrências empatar, escolheremos o valor que simplifica a próxima etapa: minimização.

### 3.2.3 Minimização

[Descrever algoritmo ISI](#)

## 4 Metodologia

O método proposto é baseado em operadores morfológicos automaticamente gerados, ou seja, construímos um segmentador genérico a partir de alguns exemplos de segmentação (pares de imagens). A seguir detalhamos cada uma das etapas envolvidas.

1. Preparação das imagens de treinamento
2. Construção dos operadores
3. Aplicação dos operadores
4. Consensualização

### 4.1 Preparação das imagens de treinamento

O primeiro passo consiste na produção dos pares de imagens de treinamento. Estes pares consistem da imagem original binarizada e de sua variante segmentada.

Para cada imagem original geramos  $n$  pares de exemplo, sendo  $n$  o número de tipos de regiões que desejamos segmentar. No caso deste trabalho nos limitamos a três: texto, título e citação.

A variante segmentada pode ser obtida através de duas estratégias: preservação ou preenchimento da região de interesse. A imagem \*exemplos de treinamento\* ilustra a geração das imagens de exemplos usando as duas estratégias.

[\\*exemplos de treinamento\\*](#)

[Falar sobre binarização em uma seção sobre o dataset e pré-processamentos](#)

## 4.2 Construção dos operadores

O algoritmo gerador de operadores morfológicos recebe como entrada um conjunto de pares de imagens de exemplo do passo anterior. Treinamos um operador para cada tipo de região utilizando a biblioteca TRIOS [ref](#). Os parâmetros para o treinamento foram ajustados experimentalmente [ref experimentos](#).

[Imagem com 2 a 3 pares de exemplos de segmentação de texto para treinamento de um operador](#)

## 4.3 Aplicação dos operadores

Construídos os operadores para cada tipo de região, aplicamos todos os operadores às imagens que desejamos segmentar obtendo um resultado para cada operador. Sendo  $n$  operadores e  $m$  imagens, obtemos  $nm$  resultados.

[exemplo de aplicação de  \$n\$  operadores](#)

## 4.4 Consensualização

A aplicação de operadores diferentes à mesma imagem pode gerar resultados incoerentes. Pixels classificados como pertencentes a mais de uma região fazem com que a união dos resultados seja impossível sem nenhum tipo de processamento. Para concluirmos a segmentação é necessário chegar a um consenso sobre qual operador possui a maior probabilidade de estar certo a cada pixel de classificação conflitante.

Partindo da observação de que pixel pertencente a uma certa região costuma estar cercados por pixels da mesma região, aplicamos um processo de escolha por maior contagem de pixels na vizinhança. Ou seja, se houver um conflito de classificação de um pixel entre texto ou título, conta-se quantos pixels na vizinhança pertencem a uma dada região e a que obtiver maior contagem ganha.

[Exemplo de conflito e contagem para consenso](#)

# 5 Experimentos

Realizamos experimentos variando diversos parâmetros em diferentes etapas da segmentação. Apesar de usarmos uma técnica de aprendizado computacional para automatizar parte do processo, identificar quais parâmetros produzem o melhor resultado a um custo desejável (tempo de processamento e poder computacional) é uma tarefa ainda experimental.

A qualidade da solução e o tempo levado para processá-lo foram medidos a fim de identificar o [italico](#) *lucus optimus*, ou seja, a combinação que produz um bom resultado com um tempo de processamento razoável.

Os parâmetros escolhidos para análise foram:

- Mistura de publicações no conjunto de treinamento.
- Quantidade de imagens no conjunto de treinamento.
- Tipos de regiões.
- Formatos de janelas.
- Tamanhos de janelas para consensualização.

A seguir detalhamos os valores dos parâmetros e os motivos de sua escolha.



## 5.1 Base de dados

As imagens utilizadas nos experimentos foram obtidas de um banco de dados construído pelos pesquisadores do PRImA ao longo de anos [ref](#). Ele inclui um conjunto de documentos que busca simular um cenário realístico de aplicação, com layouts complexos e diferentes tipos de fontes e formatos de regiões. Isto é importante para avaliar a aplicabilidade do método em situações práticas, onde um controle sobre o formato do conteúdo seria indesejável ou inviável.

No artigo [A Realistic Dataset for Performance Evaluation of Document Layout Analysis] de 2009, os autores apresentam um conjunto de dados com páginas de revistas, artigos científicos diversos, documentos modernos e não apenas históricos.

O conjunto de dados contém não só imagens mas também arquivos XML [The PAGE (Page Analysis and Ground-truth Elements) Format Framework] com metadados como informações bibliográficas (título, autor, publicação), informações das imagens (resolução, bit depth, modelo do scanner), características do layout (número de colunas, variedade de tamanhos de fontes) e informações administrativas (direitos autorais).

Os documentos são digitalizados com um cartão escuro por trás para minimizar a exposição da contra página. Posteriormente um algoritmo analisa possíveis falhas, como rotação do documento, marcando-os para redigitalização. Uma correção automática não é utilizada pois isto pode comprometer a qualidade da imagem.

Uma vez que a imagem foi aceita no banco de dados, inicia-se um processo manual de marcação do ground-truth. Este trabalho deve ser realizado da forma mais precisa possível pois é a base para determinar a corretude dos algoritmos segmentadores. Por se tratar de uma etapa muito custosa, uma ferramenta semi-automática chamada Aletheia [ref](#) é utilizada para agilizar o processo. Esta ferramenta permite a uma pessoa desenhar uma região poligonal em torno de uma região de interesse. Em seguida esta região é automaticamente ajustada pelo software, como se a pessoa estivesse colocando um elástico que aperta a região.

## 5.2 Mistura de publicações no conjunto de treinamento

Diferentes publicações trabalham com fontes e grafismos próprios. A diferença fica evidente ao comparar um jornal antigo com uma revista sobre tecnologia. Incluir imagens de publicações diversas no mesmo conjunto de imagens de treinamento pode produzir operadores menos precisos. Mesmo entre publicações do mesmo período, podemos ter diferentes tamanhos de fontes e famílias de tipos. Este experimento tem por objetivo analisar o impacto na qualidade por tentar criar operadores mais genéricos.

Testamos os seguintes conjuntos de imagens de treinamento:

- Communications of the ACM: 263, 674, 677, 680, 683, 685, 686, 689, 692, 695, 802 e 803.
- TIME Magazine: 232, 720, 721, 723, 782, 783, 784, 785 e 786.
- Publicações misturadas (inclui imagens da Fortune, Business Week, Time, CACM, IEEE e CCEJ): 158,171,232,263,701,722,235,190,720,674,792,730

[mostra imagens de revistas com fontes bem diferentes](#)

## 5.3 Tipos de regiões

Os gabaritos do conjunto de dados utilizados nestes experimentos rotulam com detalhes cada região das imagens:

1. Text

- (a) Header
  - (b) Headings (títulos)
  - (c) Capital (letras maiores)
  - (d) Drop Capital
  - (e) Credit
  - (f) Paragraph
  - (g) Floating
  - (h) Page number
  - (i) Footer
2. Graphic
  3. Math
  4. Chart
  5. Image
  6. Noise
  7. Separator
  8. Table

#### esquema de o que cada região rotula

Nos experimentos realizados nos limitamos a Text Heading e Text Paragraph por limitações de custo e tempo.

### 5.4 Quantidade de imagens de treinamento

A quantidade de imagens no conjunto de treinamento pode afetar a variedade e quantidade de padrões amostrados, influenciando diretamente a qualidade da solução. A sensibilidade a este fator pode depender do tipo de região a ser segmentada, caso a quantidade de amostras por imagem varie por região. O tempo para se treinar o operador também é impactado, pois deve-se coletar mais amostras.

Treinamos operadores utilizando cinco tamanhos de conjuntos (de uma a cinco imagens).

### 5.5 Formatos de janelas

Variando o tamanho da janela procuramos capturar padrões característicos de cada tipo de região. O tamanho da janela impacta o tempo de processamento da etapa mais custosa que é a minimização, logo descobrir um tamanho de janela com um bom custo benefício é essencial para aplicações práticas. Sabemos que nem sempre janelas maiores produzem resultados melhores [ref](#), portanto este experimento procura descobrir o melhor.

Utilizamos janelas densas e esparsas, ou seja, janelas com todos os pontos preenchidos ou apenas alguns. Os tamanhos variam de 3x3 a 7x7 para densas e de 3x3 a 11x11 para as esparsas.

#### imagens de algumas janelas

### 5.6 Tamanho da janela de consensualização

Experimentamos janelas variando de 3x3 a 15x15. Procuramos entender até quando a hipótese de que pixels de uma determinada região não ocorrem isoladamente é verdadeira.

## 5.7 Aplicação

Construímos mais de 150 operadores e os aplicamos aos nossos conjuntos de imagens de teste, cada um contendo 5 das imagens não envolvidas no treinamento dos operadores. Salvamos todas as imagens produzidas, totalizando **x GB** em resultados.

Todos os experimentos foram realizados utilizando scripts ruby para automatizar a ardua tarefa. Abaixo listamos todos os scripts com uma breve descrição.

- `sample_generator.rb` produz todos os conjuntos de imagens para treinamento e teste, ou seja, ele transforma todas as imagens e XMLs da base de dados original em entradas apropriadas para as ferramentas que utilizamos. **enxer linguagem falando como implementamos**
- `build_operators.rb` gera todos os **X** operadores utilizados nos experimentos. **falar q usa o `trios_build`**
- `apply_all.rb` aplica todos os operadores e salva os resultados, totalizando **X** imagens em **x gb**.
- `analyze.rb` extrai todas as estatísticas apresentadas na próxima seção.

## 6 Resultados

A grande quantidade de dados gerados apresenta um desafio a análise. Visualizar os resultados de uma perspectiva ou outra pode contribuir ou não para o entendimento dos resultados. Optamos por incluir tabelas com os valores medidos e alguns gráficos mais importantes que fornecem insights sobre o problema.

A métrica mais comumente utilizada nas referências teóricas sobre construção de operadores morfológicos é o MAE **ref fórmula** (sigla em inglês para Erro Absoluto Médio), porém esta métrica não é muito elucidadora. Se, por exemplo, um dado operador ideal afetar 5% dos pixels de uma imagem e o operador gerado afetar 5% dos pixels da imagem não pertencentes aos 5% do ideal, o MAE seria de 5%. Ou seja, o operador errou todos os pixels que ele deveria ter modificado e ainda modificou outros indevidamente. Ela não distingue entre diferentes tipos de erro: rotulação parcial de regiões e rotulação indevida de regiões.

Por este motivo adotamos a F-measure **ref para fórmula**, métrica mais comumente utilizada por pesquisadores de data mining.

**fórmula do f-measure**

Esta métrica também nos permite comparar o nosso método com outros trabalhos da área de segmentação de imagem que o utilizam **ref para historical bla bla bla**.

**tempo para construção dos operadores**

Tabela 2: Média da precisão na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.6955	0.8419	0.8850	0.9166	0.9396	0.6953	0.8718	0.9038	0.9527	0.9230
20%	0.8378	0.8557	0.8845	0.8926	0.9076	0.8376	0.8744	0.8859	0.9296	0.9029
30%	0.8383	0.8690	0.8911	0.8979	0.9141	0.8376	0.8751	0.8921	0.9345	0.9047
40%	0.8428	0.8635	0.8885	0.8943	0.9117	0.8376	0.8651	0.8820	0.9291	0.9025
50%	0.8418	0.8465	0.8826	0.8918	0.9125	0.8376	0.8601	0.8831	0.9289	0.9044

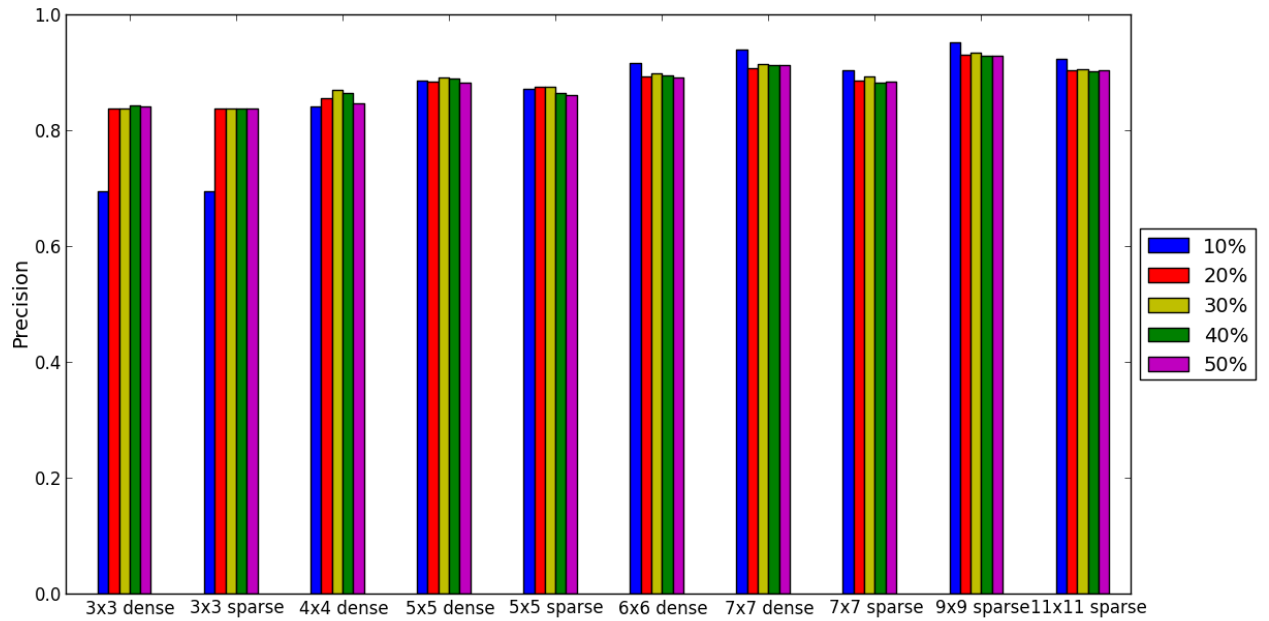


Figura 2: CACM: Classificação de parágrafos

Tabela 3: Média recall na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.9999	0.9537	0.9817	0.9736	0.9718	1.0000	0.9906	0.9937	0.9863	0.9866
20%	0.7796	0.9461	0.9778	0.9686	0.9677	0.7781	0.9894	0.9944	0.9823	0.9829
30%	0.7795	0.9316	0.9740	0.9683	0.9668	0.7781	0.9886	0.9928	0.9829	0.9848
40%	0.7789	0.9433	0.9739	0.9643	0.9641	0.7781	0.9886	0.9928	0.9810	0.9832
50%	0.7796	0.9591	0.9879	0.9802	0.9781	0.7781	0.9939	0.9953	0.9884	0.9896

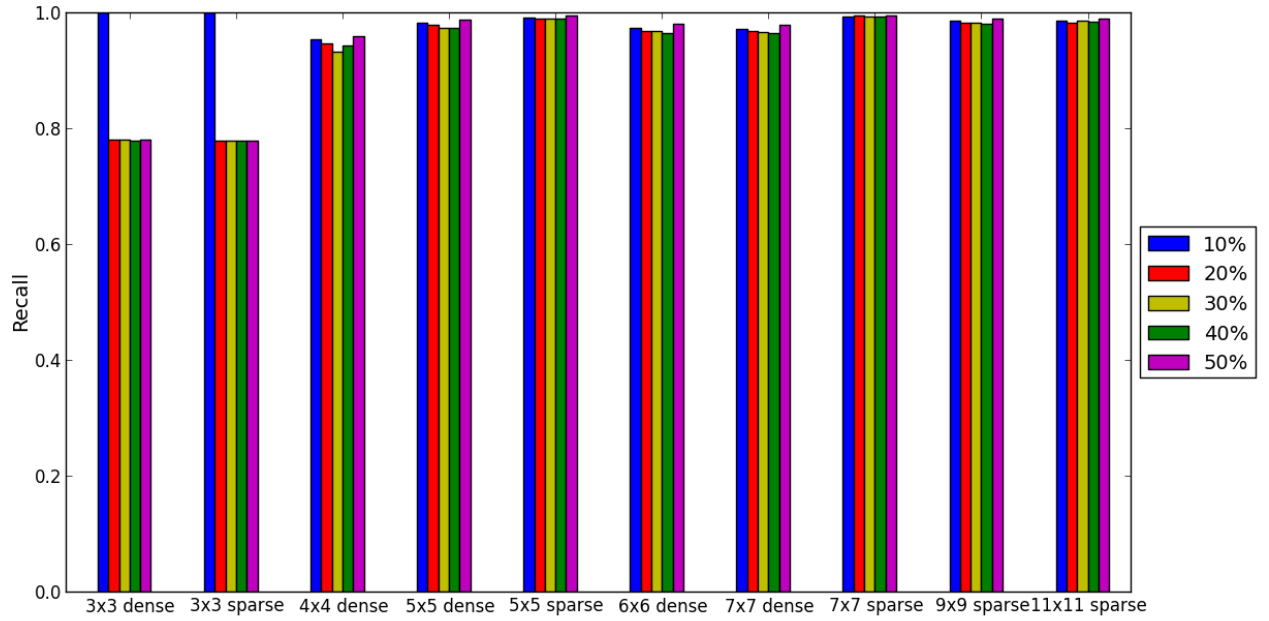


Figura 3: CACM: Classificação de parágrafos

Tabela 4: Média F1 na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.8088	0.8932	0.9304	0.9440	0.9553	0.8087	0.9268	0.9462	0.9691	0.9534
20%	0.8067	0.8979	0.9284	0.9287	0.9365	0.8058	0.9278	0.9366	0.9550	0.9409
30%	0.8068	0.8987	0.9303	0.9315	0.9396	0.8058	0.9279	0.9394	0.9580	0.9428
40%	0.8087	0.9010	0.9288	0.9277	0.9370	0.8058	0.9221	0.9337	0.9542	0.9409
50%	0.8086	0.8984	0.9319	0.9336	0.9440	0.8058	0.9215	0.9354	0.9576	0.9448

Tabela 5: Média MCC na classificação de parágrafos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0212	0.5591	0.7194	0.7746	0.8199	0.0006	0.6999	0.7775	0.8669	0.8100
20%	0.3717	0.5825	0.7123	0.7134	0.7460	0.3691	0.7025	0.7312	0.8123	0.7560
30%	0.3726	0.5940	0.7205	0.7276	0.7621	0.3691	0.7022	0.7434	0.8261	0.7671
40%	0.3815	0.5950	0.7105	0.7087	0.7497	0.3691	0.6803	0.7187	0.8095	0.7568
50%	0.3795	0.5740	0.7246	0.7302	0.7767	0.3691	0.6810	0.7259	0.8235	0.7743

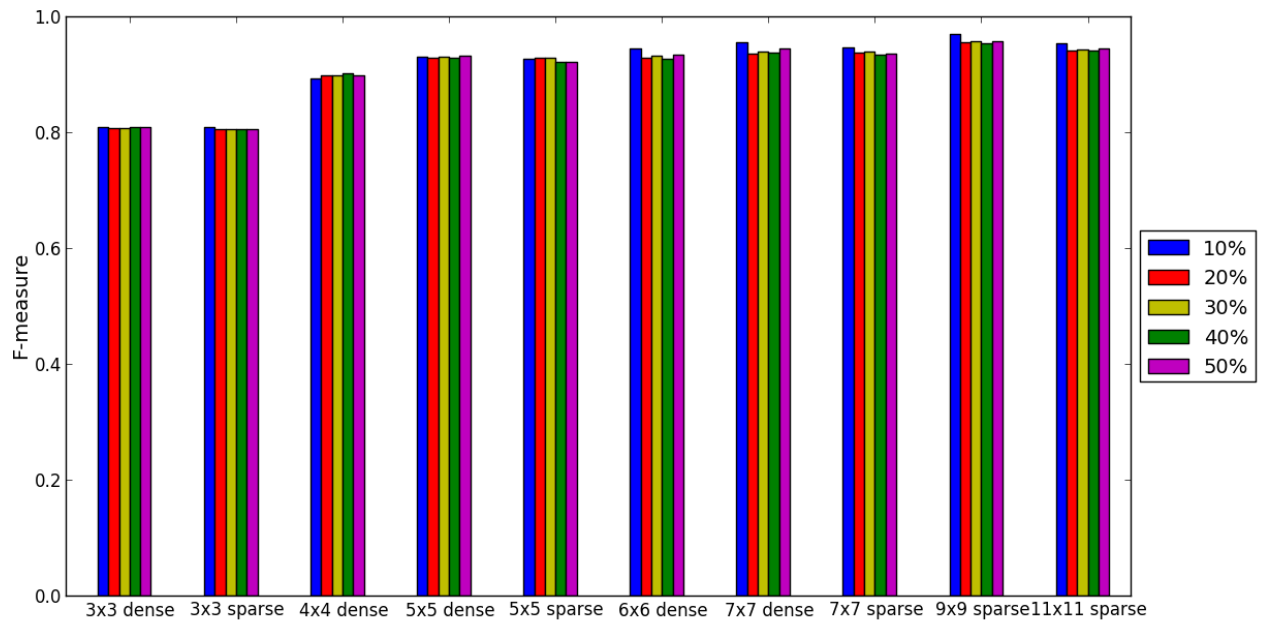


Figura 4: CACM: Classificação de parágrafos

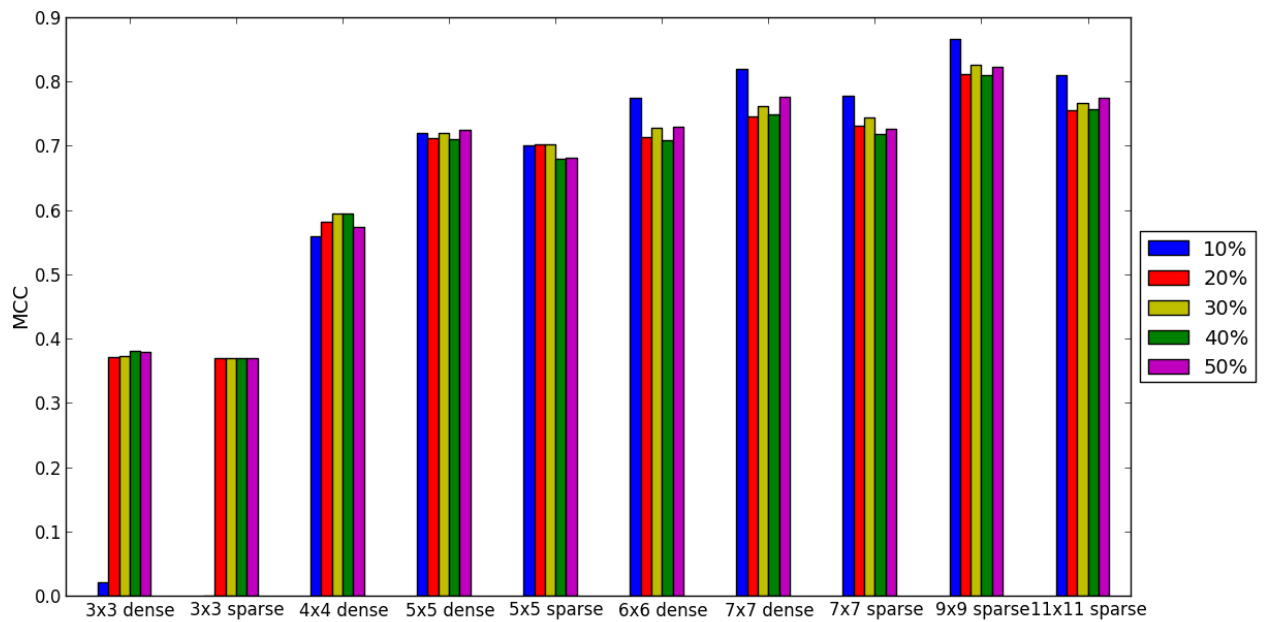


Figura 5: CACM: Classificação de parágrafos

Tabela 6: Média da precisão na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.7725	0.8186	0.8662	0.8856	0.9093	0.7226	0.8536	0.8119	0.8968	0.8095
30%	0.7729	0.8123	0.8699	0.9007	0.9177	0.7226	0.8440	0.8188	0.9032	0.8210
40%	0.7466	0.7997	0.8655	0.9008	0.9211	0.7077	0.7858	0.8155	0.9044	0.8271
50%	0.7493	0.8139	0.8775	0.9087	0.9304	0.7226	0.8364	0.8221	0.9094	0.8272

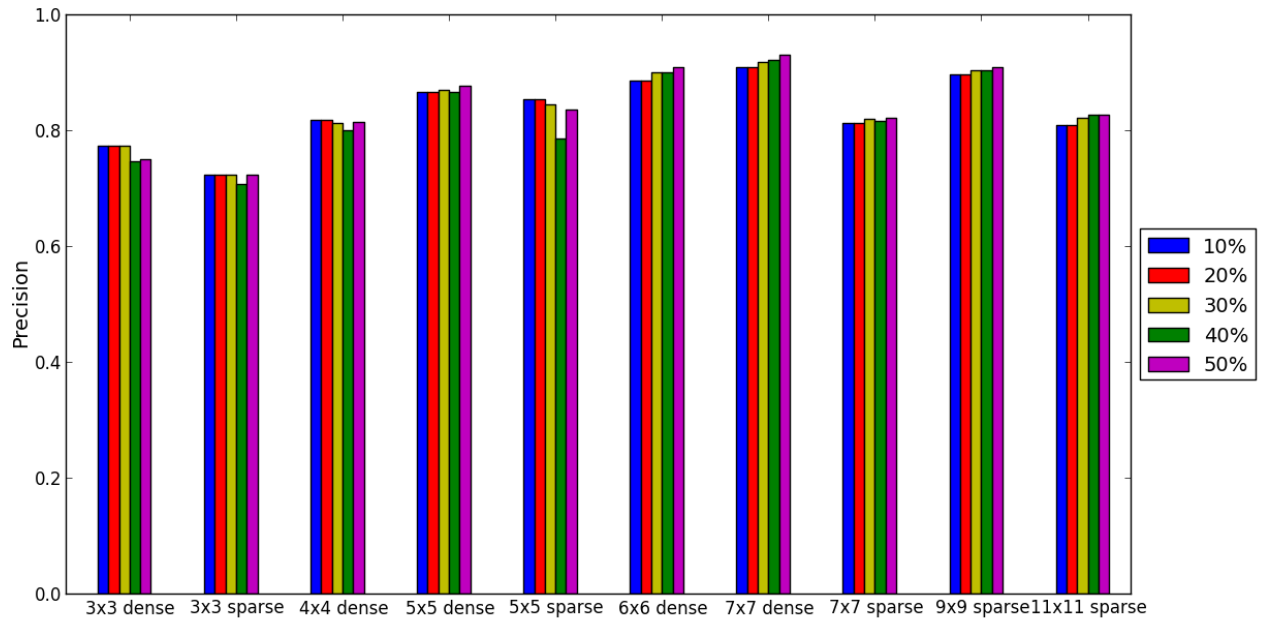


Figura 6: TIME: Classificação de parágrafos

Tabela 7: Média recall na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.6025	0.7503	0.7864	0.7966	0.8039	0.7645	0.6514	0.9461	0.8761	0.9172
30%	0.6011	0.7712	0.7990	0.8170	0.8286	0.7645	0.6825	0.9489	0.8939	0.9263
40%	0.7304	0.8761	0.9166	0.9281	0.9269	0.8087	0.9228	0.9872	0.9640	0.9734
50%	0.7212	0.8436	0.8982	0.9177	0.9251	0.7645	0.7446	0.9790	0.9595	0.9728

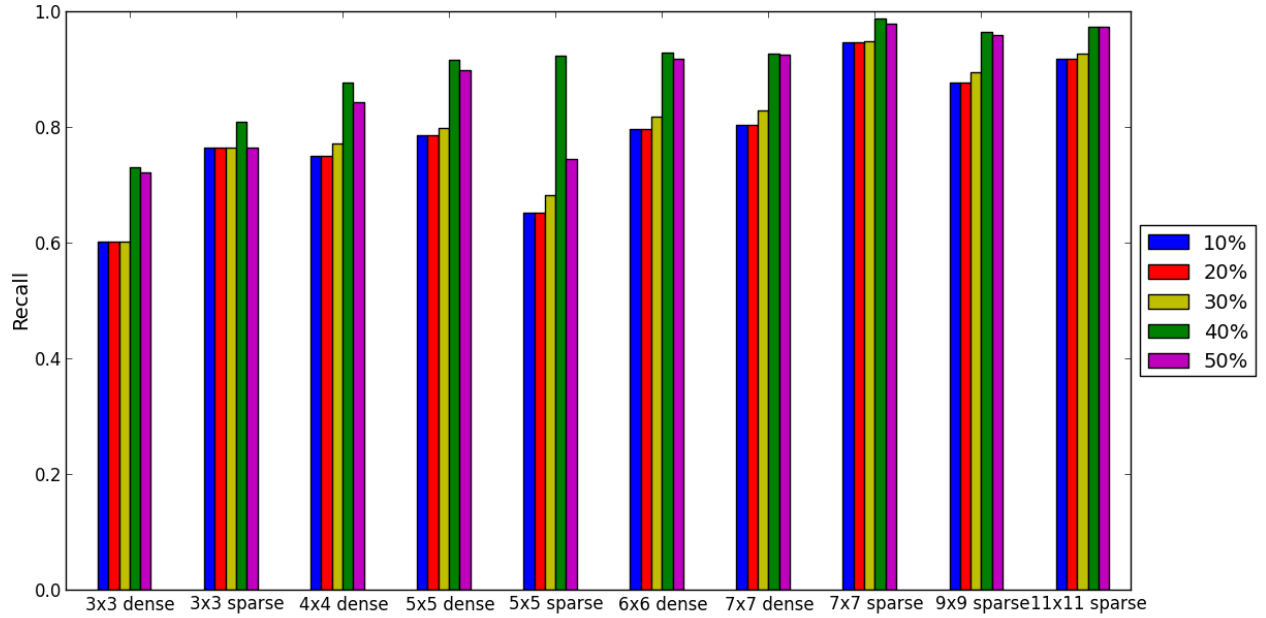


Figura 7: TIME: Classificação de parágrafos

Tabela 8: Média F1 na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.6757	0.7819	0.8230	0.8374	0.8519	0.7418	0.7373	0.8733	0.8857	0.8594
30%	0.6750	0.7902	0.8315	0.8553	0.8694	0.7418	0.7532	0.8785	0.8980	0.8699
40%	0.7372	0.8353	0.8897	0.9138	0.9235	0.7536	0.8479	0.8927	0.9330	0.8940
50%	0.7338	0.8276	0.8868	0.9123	0.9270	0.7418	0.7869	0.8932	0.9335	0.8937



## 7 Conclusão

## 8 Apêndice

### 8.1 TRIOS

O TRIOS é apenas uma implementação do processo de treinamento de operadores morfológicos. Assim, se for para colocar isso na monografia, acho que deveria estar no Apêndice.

- Imagem enquanto conjunto ou função
- Transformação entre conjuntos
- Exemplos: dilatação, erosão, abertura, fechamento, gradiente, hit-or-miss, sup-gerador.
- Operadores invariantes por translação e localmente definidos.
- W-Operadores.
- Teorema da decomposição canônica (não sei quanto disto eu consigo explicar).

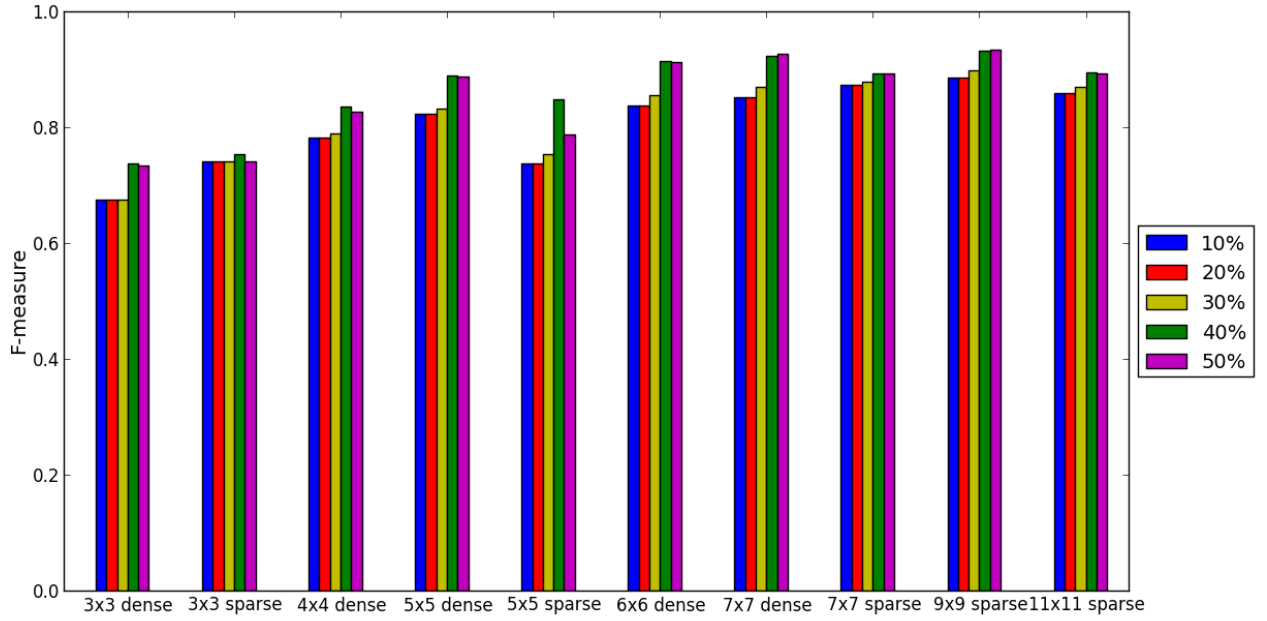


Figura 8: TIME: Classificação de parágrafos

Tabela 9: Média MCC na classificação de parágrafos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.5438	0.6720	0.7350	0.7575	0.7817	0.6004	0.6376	0.8007	0.8224	0.7784
30%	0.5431	0.6801	0.7467	0.7844	0.8059	0.6004	0.6507	0.8091	0.8407	0.7958
40%	0.6010	0.7393	0.8253	0.8642	0.8799	0.6137	0.7601	0.8352	0.8939	0.8362
50%	0.5977	0.7310	0.8227	0.8629	0.8863	0.6004	0.6854	0.8350	0.8947	0.8363

Tabela 10: Média da precisão na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.2115	0.2198	0.2301	0.2335	0.0467	0.2413	0.0910	0.2802	0.1275
20%	0.0000	0.1650	0.1894	0.2171	0.2609	0.0467	0.2485	0.0878	0.2943	0.1676
30%	0.0000	0.1352	0.1115	0.2115	0.2642	0.0467	0.2000	0.0725	0.3060	0.1495
40%	0.0000	0.1518	0.1381	0.2146	0.2637	0.0467	0.2000	0.1006	0.2945	0.1653
50%	0.0000	0.1864	0.1848	0.2334	0.2842	0.0467	0.0000	0.1069	0.3256	0.1771

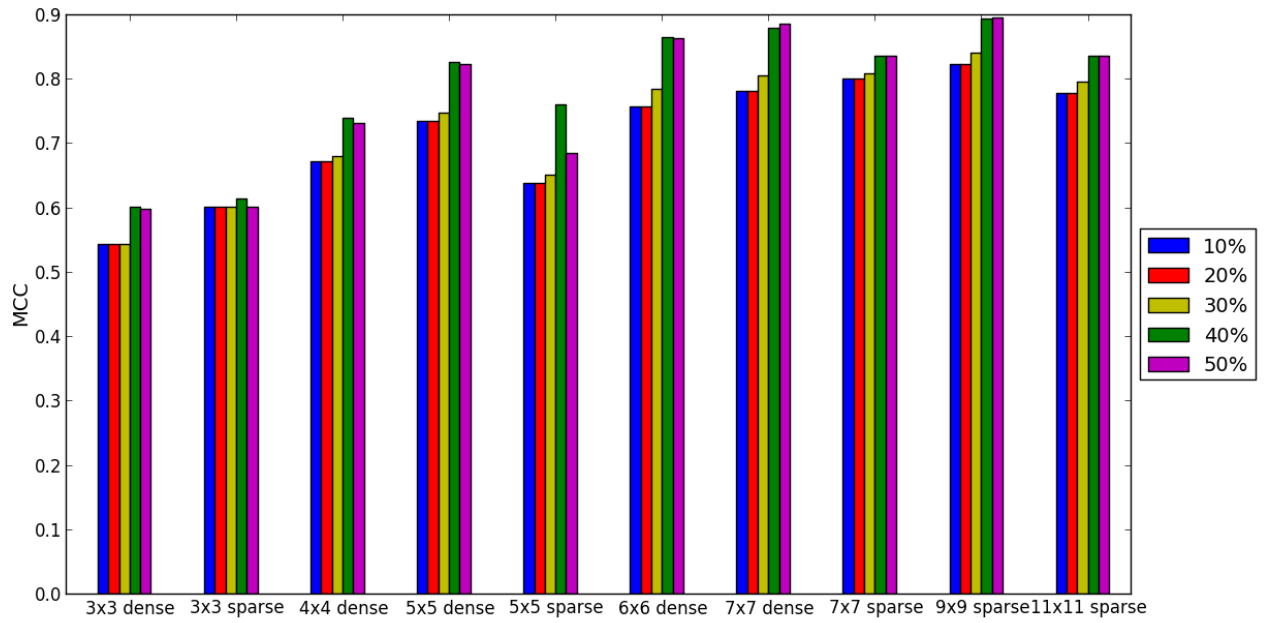


Figura 9: TIME: Classificação de parágrafos

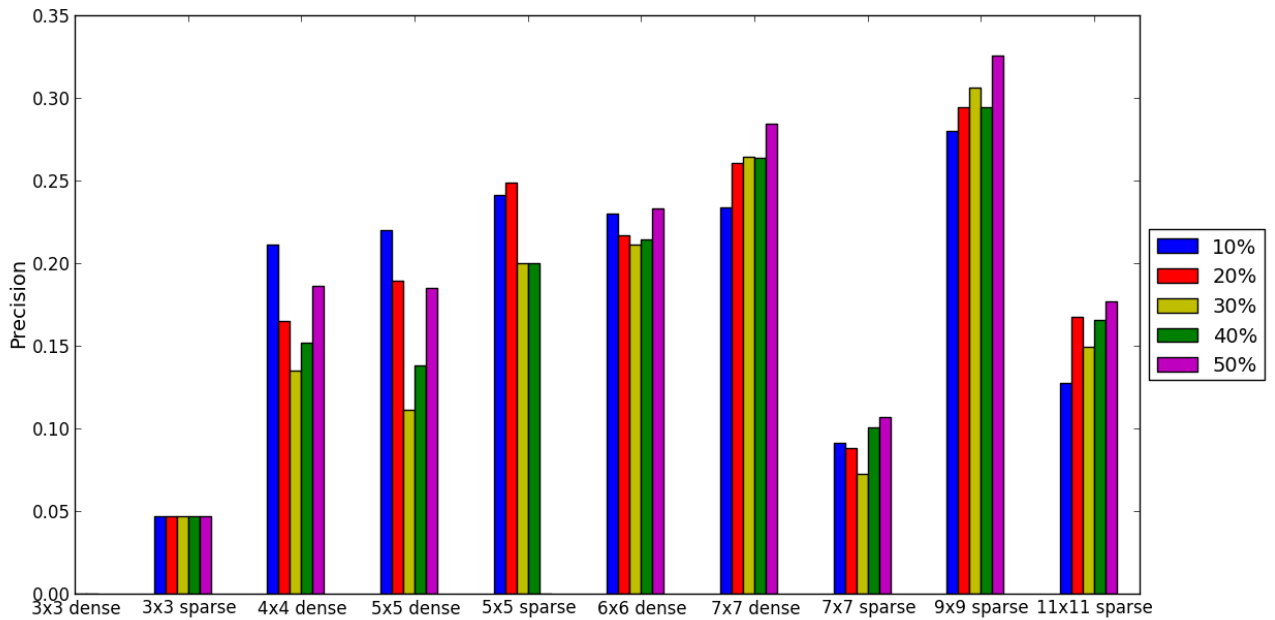


Figura 10: CACM: Classificação de títulos

Tabela 11: Média recall na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0083	0.0667	0.1420	0.2059	0.1731	0.0130	0.2912	0.2506	0.3712
20%	0.0000	0.0014	0.0382	0.1209	0.2411	0.1731	0.0028	0.2550	0.2931	0.4557
30%	0.0000	0.0003	0.0049	0.0539	0.1683	0.1731	0.0000	0.1883	0.2081	0.3980
40%	0.0000	0.0002	0.0167	0.1091	0.2598	0.1731	0.0000	0.2777	0.2925	0.4640
50%	0.0000	0.0002	0.0269	0.1113	0.2709	0.1731	0.0000	0.2866	0.3531	0.4885

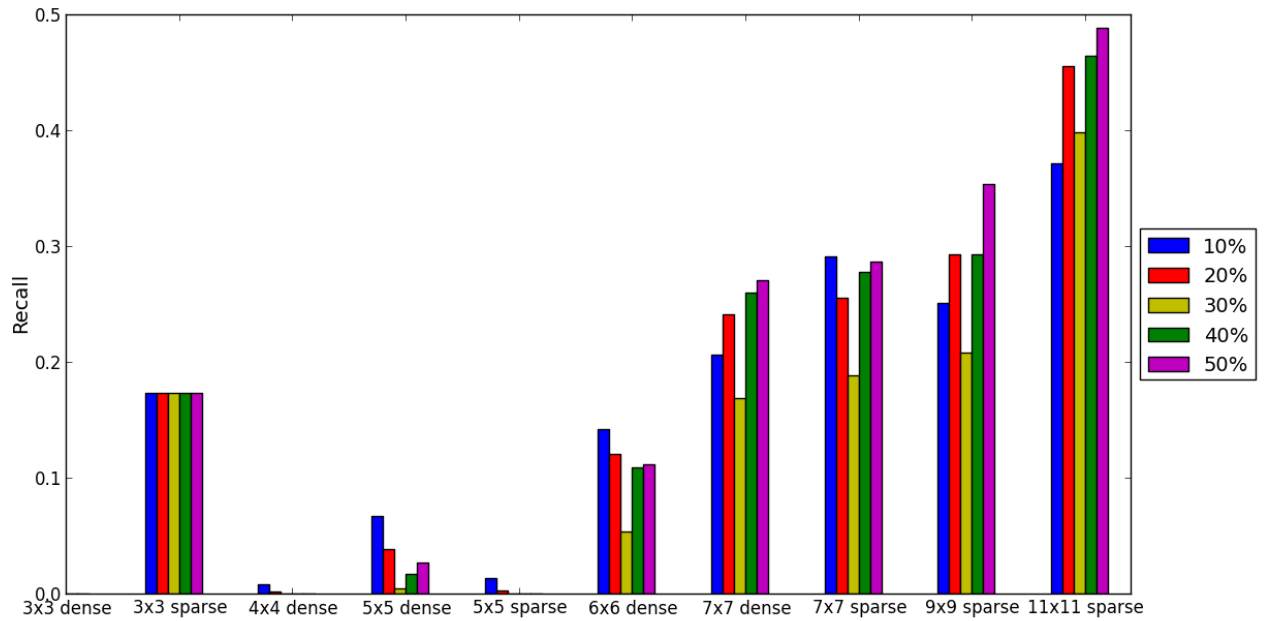


Figura 11: CACM: Classificação de títulos

Tabela 12: Média F1 na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0152	0.0869	0.1428	0.1766	0.0601	0.0234	0.1119	0.2094	0.1510
20%	0.0000	0.0028	0.0560	0.1311	0.2161	0.0601	0.0054	0.1074	0.2677	0.2116
30%	0.0000	0.0005	0.0088	0.0742	0.1799	0.0601	0.0000	0.0838	0.2308	0.1858
40%	0.0000	0.0004	0.0265	0.1252	0.2319	0.0601	0.0000	0.1234	0.2731	0.2125
50%	0.0000	0.0004	0.0429	0.1331	0.2481	0.0601	0.0000	0.1302	0.3169	0.2259

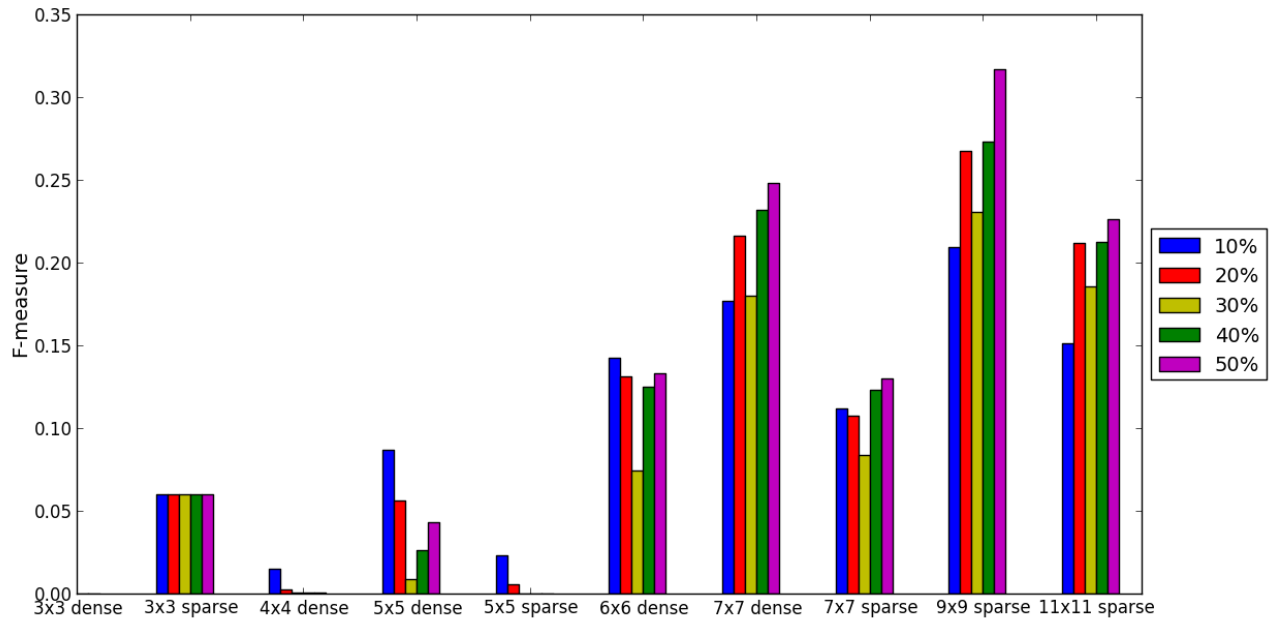


Figura 12: CACM: Classificação de títulos

Tabela 13: Média MCC na classificação de títulos do conjunto de dados CACM

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0243	0.0739	0.1184	0.1484	-0.0879	0.0357	0.0239	0.1958	0.0964
20%	0.0000	0.0078	0.0508	0.1065	0.1849	-0.0879	0.0176	0.0117	0.2342	0.1640
30%	0.0000	0.0020	0.0058	0.0676	0.1551	-0.0879	0.0006	-0.0159	0.2023	0.1264
40%	0.0000	0.0024	0.0203	0.1003	0.1975	-0.0879	0.0008	0.0319	0.2372	0.1637
50%	0.0000	0.0033	0.0416	0.1115	0.2162	-0.0879	0.0000	0.0435	0.2838	0.1851

Tabela 14: Média da precisão na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0000	0.0020	0.0571	0.0776	0.0189	0.0000	0.0232	0.0510	0.0277
30%	0.0000	0.0000	0.0018	0.0063	0.0137	0.0189	0.0000	0.0227	0.0151	0.0217
40%	0.0000	0.0000	0.0081	0.0203	0.0447	0.0189	0.0000	0.0271	0.0600	0.0297
50%	0.0000	0.0147	0.0122	0.0495	0.0721	0.0189	0.0000	0.0274	0.1172	0.0341

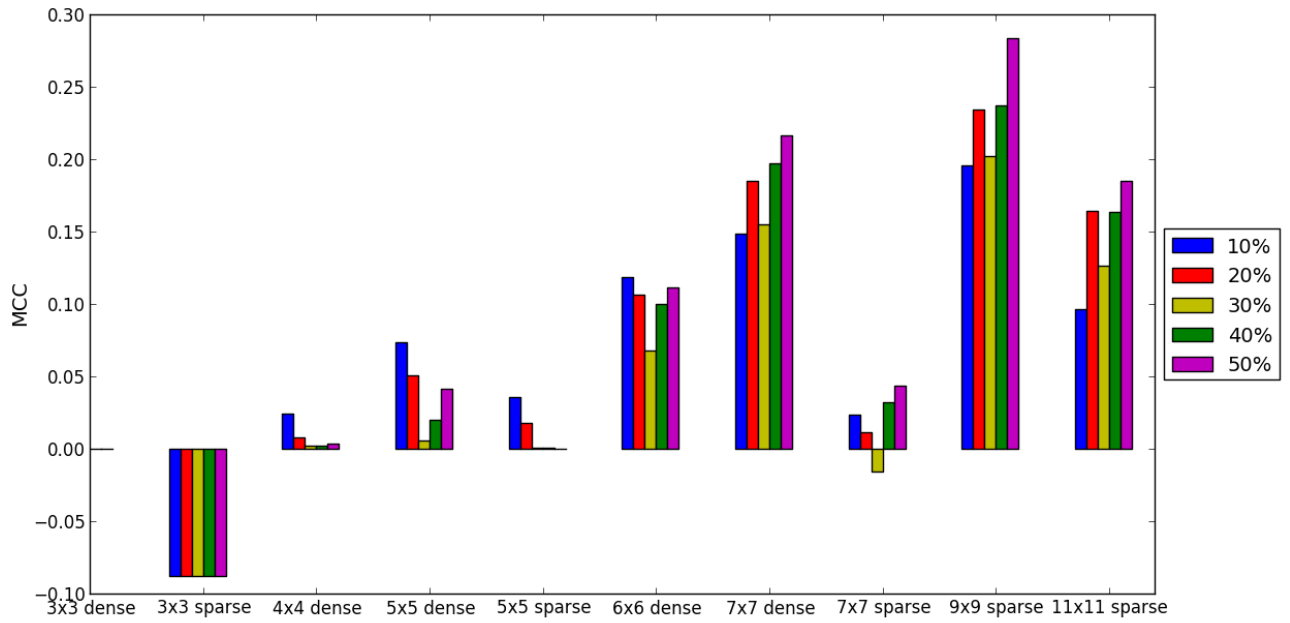


Figura 13: CACM: Classificação de títulos

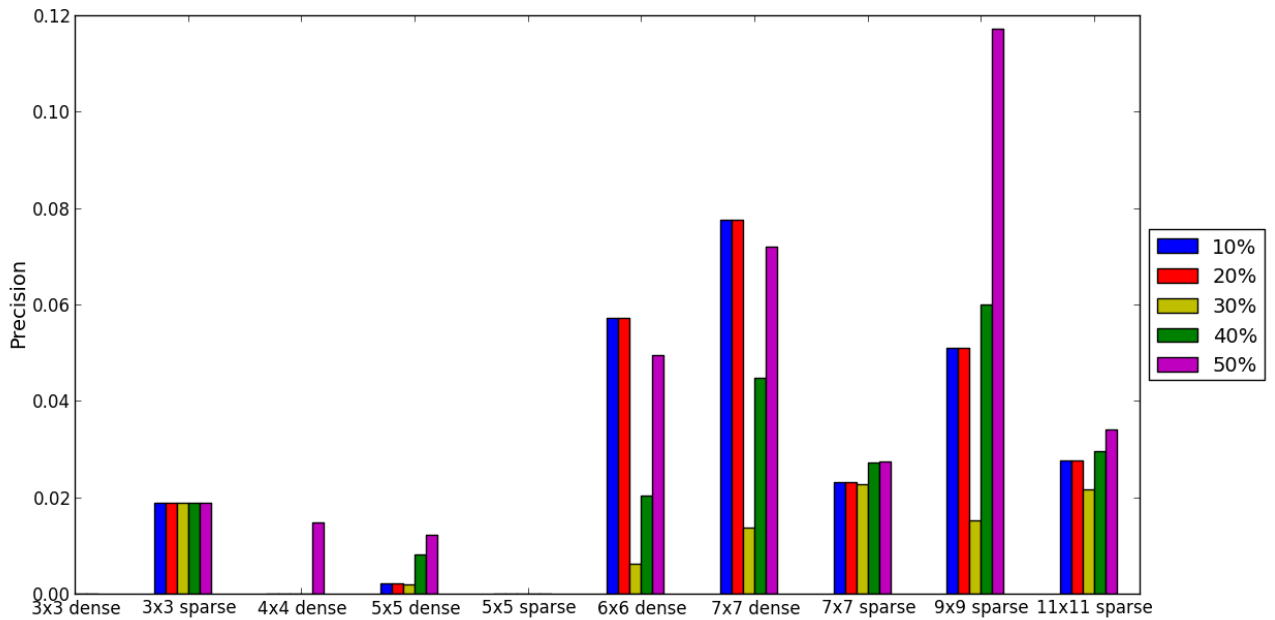


Figura 14: TIME: Classificação de títulos

Tabela 15: Média recall na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0000	0.0007	0.0610	0.1410	0.1786	0.0000	0.1600	0.0451	0.2089
30%	0.0000	0.0000	0.0004	0.0053	0.0166	0.1786	0.0000	0.1558	0.0108	0.1598
40%	0.0000	0.0000	0.0001	0.0046	0.0210	0.1786	0.0000	0.1626	0.0138	0.1821
50%	0.0000	0.0000	0.0003	0.0056	0.0256	0.1786	0.0000	0.1607	0.0111	0.1943

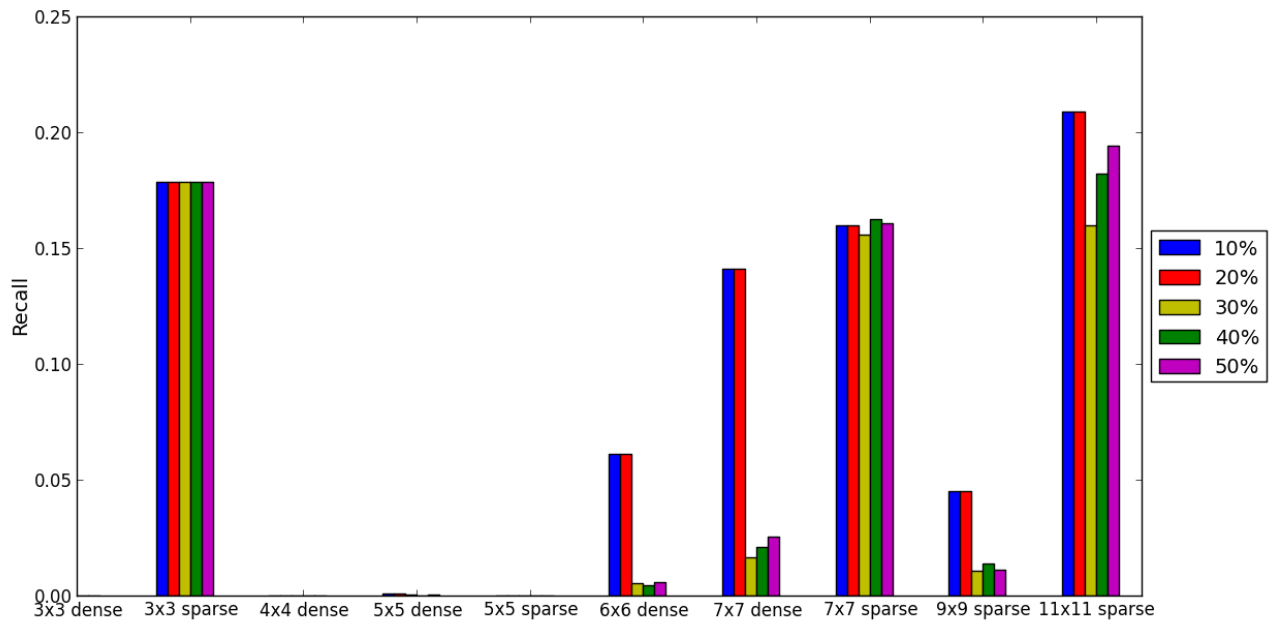


Figura 15: TIME: Classificação de títulos

Tabela 16: Média F1 na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	0.0000	0.0010	0.0568	0.0966	0.0331	0.0000	0.0392	0.0451	0.0476
30%	0.0000	0.0000	0.0007	0.0056	0.0146	0.0331	0.0000	0.0383	0.0123	0.0371
40%	0.0000	0.0000	0.0003	0.0073	0.0272	0.0331	0.0000	0.0449	0.0214	0.0494
50%	0.0000	0.0000	0.0005	0.0099	0.0367	0.0331	0.0000	0.0452	0.0199	0.0562

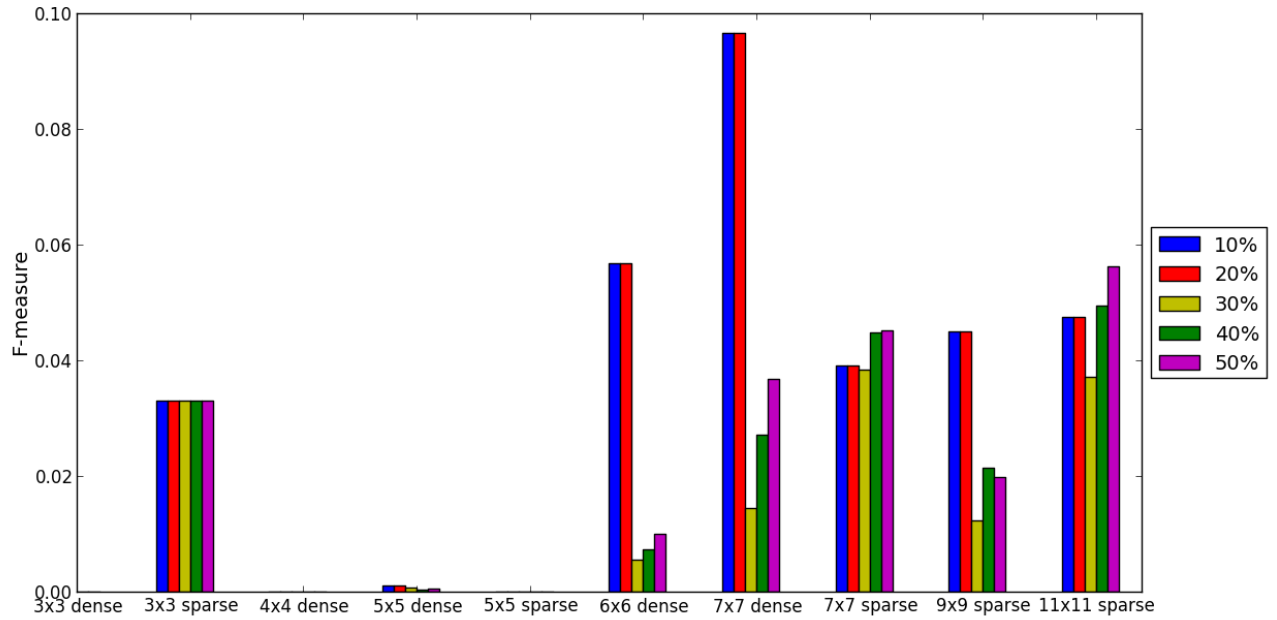


Figura 16: TIME: Classificação de títulos

Tabela 17: Média MCC na classificação de títulos do conjunto de dados TIME

	janelas									
	densa					esparsa				
	3x3	4x4	5x5	6x6	7x7	3x3	5x5	7x7	9x9	11x11
10%	0.0000	-0.0015	-0.0147	0.0281	0.0649	-0.0371	0.0000	-0.0177	0.0202	-0.0050
30%	0.0000	-0.0014	-0.0122	-0.0186	-0.0148	-0.0371	0.0000	-0.0190	-0.0087	-0.0215
40%	0.0000	0.0000	-0.0024	-0.0027	0.0109	-0.0371	0.0000	-0.0061	0.0151	0.0018
50%	0.0000	-0.0001	-0.0015	0.0069	0.0251	-0.0371	0.0000	-0.0053	0.0263	0.0121



- Conjuntos aleatórios S e I. Caracterização por um processo estacionário local (X, y).
- Otimalidade de Psy com base num operador localmente definido (MAE).
- Algoritmo: Estimativa de  $P(y | X)$ , decisão, generalização (ISI?).
- Bias-Variance Tradeoff
- Explorando estrutura de Psy? (talvez isso caiba melhor na lista de estratégias a seguir)
- Escolha da janela ótima.
- Operador multi-nível.

## 8.2 Algoritmo de Otsu para binarização

**mover esta subseção para dentro de dataset e pré processamentos**

Implementar e avaliar cada algoritmo binarizador vai além do escopo deste trabalho, portanto escolhemos um bom algoritmo segundo os resultados obtidos em [2]. Como o próprio artigo aponta, não foi encontrado um método que apresentasse desempenho superior em todas os cenários de teste realizados.

Os requisitos para a escolha foram o da independência de supervisionamento e parametrização. Caso o algoritmo demandasse ajustes específicos de acordo com a imagem, o seu uso comprometeria a promessa de automatização. Dentre a lista de soluções com esta característica, escolhemos o algoritmo de Otsu [3], por ser de fácil implementação e apresentar resultados satisfatórios nos experimentos realizados.

O algoritmo de Otsu encontra um nível de cinza  $t$  tal que a soma ponderada da variância dentro das classes  $\mathbb{F} = \{x: f(x) \geq t\}$  (foreground) e  $\mathbb{B} = \{x: f(x) < t\}$  (background) seja minimizada, ou seja,

$$t = \operatorname{argmin}\{w_{\mathbb{B}}\sigma_{\mathbb{B}}^2 + w_{\mathbb{F}}\sigma_{\mathbb{F}}^2\} \quad (5)$$

onde  $w_{\mathbb{B}} = \frac{|\mathbb{B}|}{|E|}$  e  $w_{\mathbb{F}} = \frac{|\mathbb{F}|}{|E|}$  são os pesos respectivamente do background e foreground e  $\sigma_{\mathbb{B}}^2$  e  $\sigma_{\mathbb{F}}^2$  são as variâncias das classes.

A tabela 19 apresenta um passo a passo do algoritmo aplicado à figura 18.

Como podemos notar, para  $t = 50\%$  atingimos o menor valor de  $\sigma_w^2 \approx 512693389$ . Neste caso o limiar coincide com o único vale no histograma, porém isto nem sempre será válido. Algumas imagens não possuem vales bem definidos. Este algoritmo não se baseia no formato do histograma mas sim na coesão intra classe e na separabilidade das classes.

Uma desvantagem de utilizar este algoritmo é a influência da média de todo os pixels da imagem. Isto pode fazer com que o limiar ótimo para a página toda não seja o mesmo que o de dentro de uma janela.

## Referências

- [1] Rangachar Kasturi, Lawrence O’Gorman, and Venu Govindaraju. Document image analysis: A primer. *Sadhana*, 27(1):3–22, 2002.
- [2] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, January 2004.

Tabela 18: Imagem original em escala de cinza e correspondente binarizada com  $t = 50\%$ .

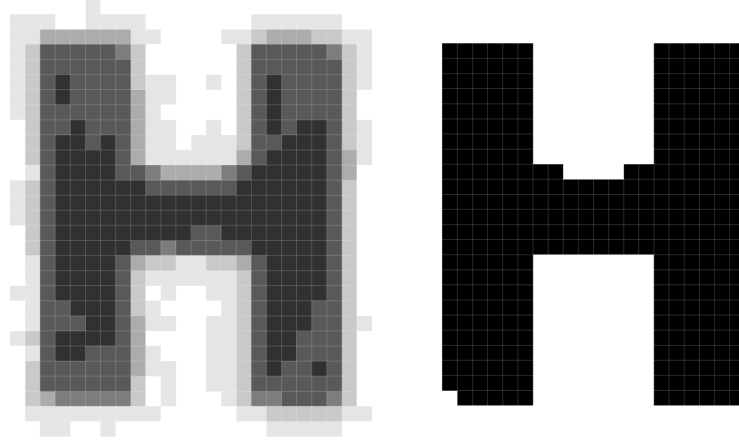
















Tabela 19: Estágios da execução do algoritmo de Otsu.

original	25%	37,5%	50%	62,5%	75%	87,5%
						
						
$w_B$	0.1931	0.4015	0.4179	0.4532	0.5479	0.6969
$\mu_B$	34.00	51.64	53.61	61.37	83.08	112.56
$\sigma_B^2$	7.27	288.58	372.94	1054.08	3128.03	5656.37
$w_F$	0.80	0.59	0.58	0.54	0.45	0.30
$\mu_F$	184.87	225.55	229.03	233.95	243.79	255.0
$\sigma_F^2$	5799.89	1409.01	1006.11	673.10	255.43	0.0
$\sigma_w^2$	21495165144	967521582	512693389	595067475	4269882800	17660993341

- [3] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.

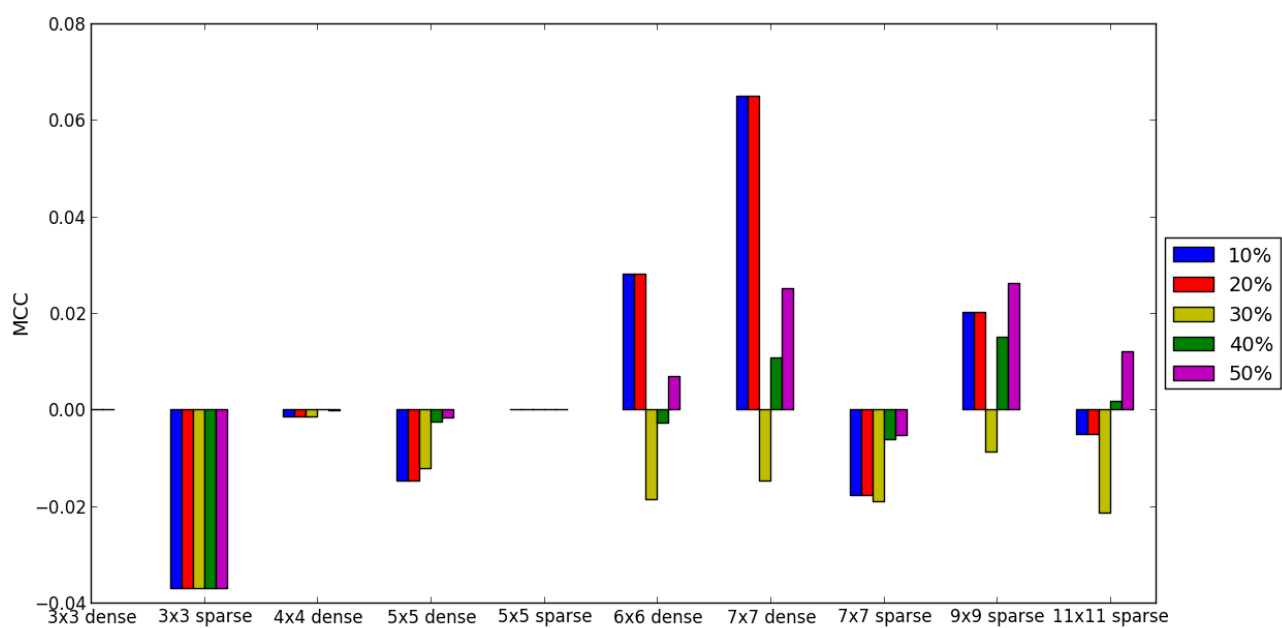


Figura 17: TIME: Classificação de títulos