

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA INFORMATICA

Titolo tesi

Relatore:

PROF. NOME COGNOME

Laureando:

NOME COGNOME

1234567

Anno Accademico 2019/2020

Abstract

Inserire abstract. I margini nell'abstract sono stati ridotti di un centimetro. In caso non si volesse questa riduzione rimuovere *changemargin*.

Indice

1	Primo Capitolo	1
1.1	Sezione 1	1
1.2	Sezione 2	1
1.2.1	Sottosezione	1
2	Immagini e Tabelle	3
2.1	Immagine singola	3
2.2	Immagine multipla	4
2.3	Tabelle	4
3	Formule	5
4	Pseudocodice e codice	7
4.1	Pseudocodice	7
4.2	Codice	7
	Bibliografia	9

Capitolo 1

Primo Capitolo

La struttura utilizzata in questo template non è obbligatoria, però ritengo che sia molto comoda per evitare di scrivere file troppo lunghi e di avere un controllo migliore sulla struttura. Questa prevede di scrivere l'introduzione al capitolo in un file salvato nella cartella principale e di sviluppare le sezioni all'interno di una cartella. Esempio di richiamo ad un riferimento [2].

«Each thing says what it is...a fruit says 'Eat me'; water says 'Drink me'; thunder says 'Fear me'...» (Koffka, 1935)

1.1 Sezione 1

Ad ogni sezione, in questo template, corrisponde un file all'interno della cartella relativa al capitolo.

1.2 Sezione 2

Una sezione può contenere una sottosezione. In questo caso, è stato deciso di non creare altri file...

1.2.1 Sottosezione

...ma di scrivere il testo all'interno del file della sezione corrente.

Capitolo 2

Immagini e Tabelle

Metodi più comuni per inserire immagini e esempi di tabelle.

2.1 Immagine singola

Per fare riferimento ad un immagine, come ad ogni altro elemento a cui viene attribuita una *label* è disponibile il comando *ref*. Riferimento a Figura 2.1.

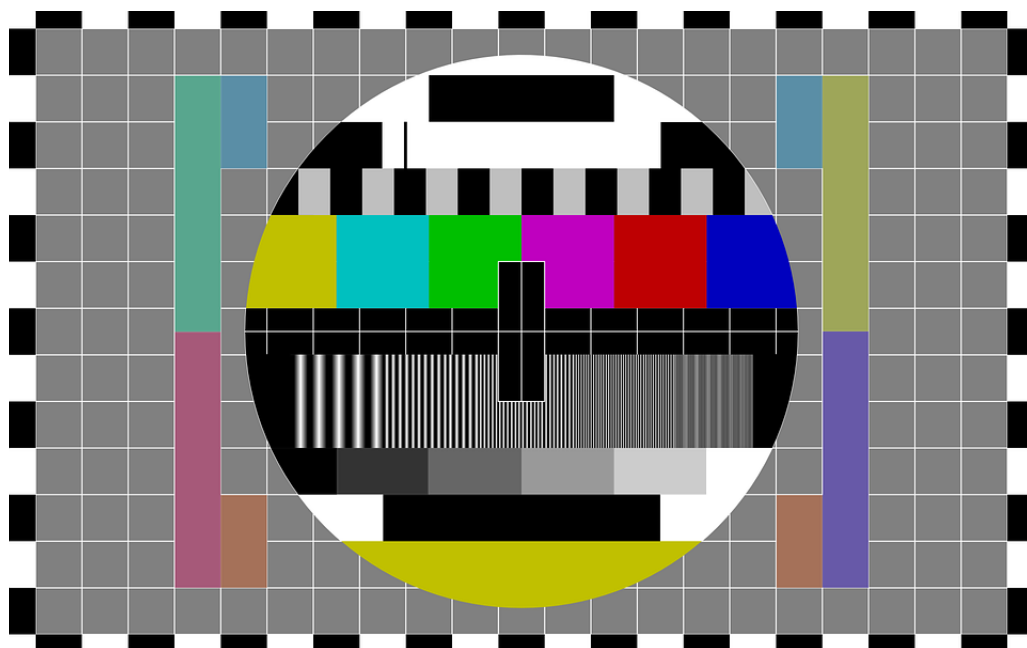


Figura 2.1: Didascalia

2.2 Immagine multipla

Inserire più “sottofigure” in una figura.

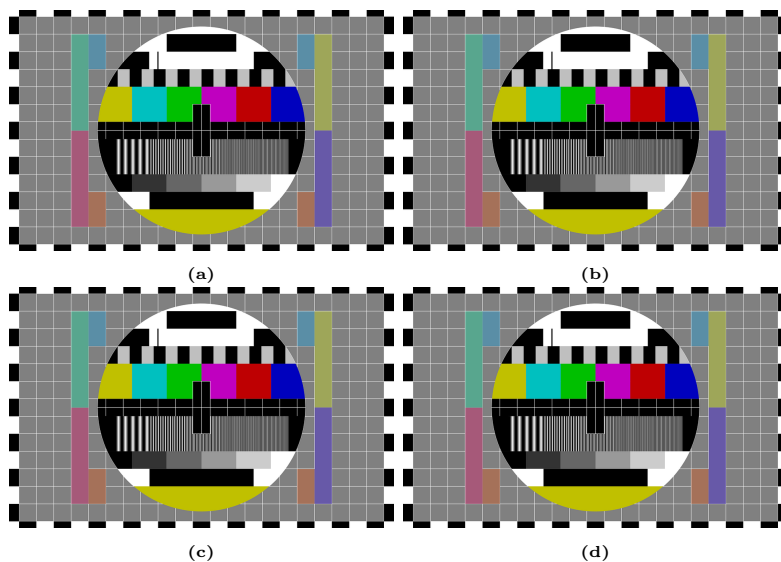


Figura 2.2: Esempio di figura composta da 4 figure.

2.3 Tabelle

Nella seguente tabella vengono mostrati alcuni esempi di separazione delle righe. La separazione delle colonne avviene all’interno delle parentesi grafe dopo il comando *tabular*.

cella1	cella2	cella3
cella4	cella5	cella6
cella7	cella8	cella9
cella10	cella11	cella12

È possibile, inoltre, fissare la dimensione delle colonne.

cella1	cella2	cella3
cella4	cella5	cella6 cella6 cella6 cella6
cella7 cella10	cella8 cella11	cella9 cella12

Capitolo 3

Formule

Per inserire delle formule matematiche è possibile utilizzare due metodi:

- in linea: inserendo la formula tra due caratteri \$.
- utilizzando l'ambiente *equation*.

Esempio di formula in linea $x(t) = x_0 + v_0t + \frac{1}{2}at^2$.

Esempio di utilizzo dell'ambiente *equation*:

$$x(t) = x_0 + v_0t + \frac{1}{2}at^2 \tag{3.1}$$

Possiamo usare le *label* anche per le equazioni. Legge oraria nell'Equazione 3.1. Infine, un esempio di formula su più righe:

$$\begin{aligned} x(t) &= x_0 + v_0t + \frac{1}{2}at^2 \\ &= x_0 + v_0t + \frac{1}{2}\frac{F}{m}t^2 \end{aligned} \tag{3.2}$$

Capitolo 4

Pseudocodice e codice

In questo template per l'inserimento di pseudocodice è stato utilizzato il pacchetto *algpseudocode*. Per quanto riguarda l'inserimento del codice è possibile utilizzare il comando *verb* per inserire in linea oppure *lstlisting* per inserire blocchi di codice.

4.1 Pseudocodice

Algorithm 1 Nome algoritmo

Input: Input dell'algoritmo

Output: Output dell'algoritmo

```
1: variabile  $\leftarrow$  assegnazione valore
2: valore_ritornato  $\leftarrow$  FUNZIONEPROVA(param1, param2)
3: for element in list do
4:   res  $\leftarrow$  DOSOMETHING(element)
5: end for
6: if condizione1 then
7:   do something
8: else if condizione2 then
9:   do something else
10: else
11:   print "Hello World"
12: end if
```

4.2 Codice

È possibile inserire codice in linea: `print("Hello World")`.

Inoltre è possibile usare l'ambiente *lstlisting* configurando il layout del blocco di

codice nel file *layout.tex*. Il codice può essere importato da un file esterno che metteremo nella cartella *code*.

```
1 # Number of trees in random forest
2 n_estimators = [int(x) for x in np.linspace(start = 100, stop =
    1000, num = 10)]
3 # Number of features to consider at every split
4 max_features = ['log2', 'sqrt']
5 # Maximum number of levels in tree
6 max_depth = [int(x) for x in np.linspace(10, 150, num = 15)]
7 max_depth.append(None)
8 # Minimum number of samples required to split a node
9 min_samples_split = [2, 5, 10]
10 # Minimum number of samples required at each leaf node
11 min_samples_leaf = [1, 2, 4, 6, 8]
12 # Method of selecting samples for training each tree
13 bootstrap = [True, False]
14 random_grid = {'n_estimators': n_estimators,
15               'max_features': max_features,
16               'max_depth': max_depth,
17               'min_samples_split': min_samples_split,
18               'min_samples_leaf': min_samples_leaf,
19               'bootstrap': bootstrap
20               }
21
22 rfc = RandomForestClassifier()
23 rfc_random = RandomizedSearchCV(estimator=rfc,
    param_distributions=random_grid, n_iter=1000, cv=5, verbose=2,
    random_state=3, n_jobs=4)
24
25 rfc_random.fit(df_downsampled, labels_downsampled)
```

Listing 4.1: Didascalia.

Bibliografia

- [1] Buitinck L., Louppe G., Blondel M., Pedregosa F., Mueller A., Grisel O., Niculae V., Prettenhofer P., Gramfort A., Grobler J., Layton R., Vanderplas J., Joly A., Holt B., Varoquaux G., *API design for machine learning software: Experiences from the scikit-learn project*, 2013.
- [2] *Convolutional Neural Networks*, <http://deeplearning.net/tutorial/lenet.html>, last consultation: 07/01/2020.