

---

# **nms Documentation**

**Tom Hoag**

**Aug 29, 2018**



---

## Contents

---

<b>1</b>	<b>Summary &amp; Notes</b>	<b>1</b>
<b>2</b>	<b>Install</b>	<b>3</b>
<b>3</b>	<b>Basic Usage</b>	<b>5</b>
<b>4</b>	<b>Most Useful Functions</b>	<b>7</b>
<b>5</b>	<b>nms</b>	<b>9</b>
5.1	nms package . . . . .	9
<b>6</b>	<b>Index</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>



---

## Summary & Notes

---

Given a list of rectangles (or rotated rectangles or polygons) and a corresponding list of scores (confidences), the Non Maximal Suppression functions below will return a list of indicies. These indicies correspond to the rectangles (or rotated rectangles or polygons) that have the highest scores and the minimum overlap with other elements in the list.

Rectangles (sometimes referred to as rects and boxes) are expressed as a 4-tuple: (x, y, w, h). Where x, y is the coordinate of the upper left hand corner of the rectangle. w and h are the width and height of the rectangle.

Rotated rectangles (sometimes referred to as rrects or rboxes) are expressed as a 5-tuple: (x, y, w, h, r). Where the first four members are as described for rectangles and r is the angle, in radians, of rotation around the center of the rectangle.

Polygons are expressed as a list of 2-tuples of (x, y). Each x, y describes a verticie of the polygon. The edges of the polygon are described by each pair of consecutive vertices. That is (x0, y0), (x1, y1) describe an edge of the polygon. The polygons are closed by the edge (xn, yn), (x0, y0).

If you only have a need for NMS on rectangles and are already using OpenCV, you may want to consider using `cv2.dnn.NMSBoxes`. Note that the input for NMSBoxes is a numpy array of ((x, y), (w, h), d) where d is the angle *in degrees* that the rectangle is rotated about its center.

nms was developed with Python 3.7 and OpenCV 4.0.0-pre

This code is beta and lacking in test coverage (read: no coverage)



## CHAPTER 2

---

### Install

---

nms has dependencies on cv2 (OpenCV 4.0.0-pre), numpy and math. [Adrian Rosenbrock has excellent instructions for installing OpenCV 4.0 on MacOS and Ubuntu](#)

Include nms in your requirements.txt or install into your environment:

```
pip install nms
```





## CHAPTER 3

---

### Basic Usage

---

Given a list of rotated rectangles *rrects* and confidences *scores*:

```
indices = nms.nms.nms_rboxes(rrects, scores)
```

By default, `nms.nms.nms_boxes()`, `nms.nms.nms_polygons()` and `nms.nms.nms_rboxes()` use the Malisiewicz et. al. NMS Algorithm. If you would prefer to use the OpenCV2 Fast algorithm or Felzenszwalb et. al. algorithm you can do so with the optional parameter *nms\_function*. For example:

```
indices = nms.nms.nms_rboxes(rrects, scores, nms_function=malisiewicz)
```

`nms.nms.nms_boxes()`, `nms.nms.nms_polygons()` and `nms.nms.nms_rboxes()` will also accept a list of key word arguments (*\*\*kwargs*). The *kwargs* are passed to the NMS function to set values for the NMS threshold, confidence threshold, *top\_k* and *eta* (*eta* is only applicable when using Fast).

For example, to set the NMS threshold to 0.5 (note that use of *kwargs* requires that *nms\_function* parameter be present):

```
indices = nms.nms.nms_rboxes(rrects, scores, nms_function=malisiewicz, nms_  
→threshold=0.5)
```

Please see the docs for `nms.fast.NMS()`, `nms.malisiewicz.NMS()` and `nms.felzenszwalb.NMS()` for additional information on *kwargs*



## CHAPTER 4

---

### Most Useful Functions

---



## 5.1 nms package

### 5.1.1 Submodules

### 5.1.2 nms.fast module

### 5.1.3 nms.felzenszwalb module

### 5.1.4 nms.helpers module

### 5.1.5 nms.malisiewicz module

### 5.1.6 nms.nms module

### 5.1.7 Module contents



## CHAPTER 6

---

Index

---





**n**

nms, 9



## N

nms (module), [9](#)