Table S1. The Landsat image numbers in different phenological periods and Landsat sensors from 1989-91 to 2019-21. TM is the Thematic Mapper Sensor, ETM+ is the Enhanced Thematic Mapper Plus sensor, OLI is the Operational Land Imager sensor, FTP is the flooding and transplanting period, and GHP is the growing and harvesting period.

Sensor	year	Numbers in FTP	Numbers in GHP
TM	1989-91	61	52
	1992-94	64	52
	1995-97	52	56
	1998-00	63	62
	2001-03	44	39
	2004-06	54	66
	2007-09	61	56
	2010-11	36	33
ETM+	2012	23	13
OLI	2013-15	69	98
	2016-18	91	96
	2019-21	109	106

Table S2. The mapping accuracy in six periods. KC is Kappa coefficient, OA is overall accuracy, PA is producer's accuracy of paddy rice, UA is user's accuracy of paddy rice, and FS is F1 score of paddy rice.

Year Periods	KC	OA	PA	UA	FS
1989-91	71.262%	83.000%	85.366%	97.222%	90.909%
1995-97	83.271%	90.333%	94.595%	92.105%	93.333%
1998-00	85.903%	92.000%	90.000%	100.000%	94.737%
2004-06	83.083%	89.333%	93.333%	95.455%	94.382%
2010-12	78.342%	88.333%	80.000%	88.889%	84.211%
2013-15	80.457%	89.324%	86.364%	82.609%	84.444%

Supplementary information A. Paddy rice mapping code in 1998-2000.

```
import os
import ee
import geemap
ee.Initialize()
Map=geemap.Map()

def cloud_landsat8(image):
    qa=image.select('QA_PIXEL')
    bitDilatedCloud=(1<<1)
    bitCirrus=(1<<2)
    bitCloud=(1<<3)
    bitCloudShadow=(1<<4)

    mask=qa.bitwiseAnd(bitDilatedCloud).eq(0).And(
        qa.bitwiseAnd(bitCirrus).eq(0)).And(
        qa.bitwiseAnd(bitCloud).eq(0)).And(
        qa.bitwiseAnd(bitCloudShadow).eq(0))

    opticalBands=image.select('SR_B.').multiply(0.0000275).add(-0.2)
    thermalBands=image.select('ST_B.*').multiply(0.00341802).add(149.0)
    return image.addBands(opticalBands,None,True)    \
                .addBands(thermalBands,None,True)    \
                .updateMask(mask)

def cloud_landsat5(image):
    qa=image.select('QA_PIXEL')
    bitCloud=(1<<3)
    bitCloudShadow=(1<<4)
    bitDilatedCloud=(1<<1)

    mask=qa.bitwiseAnd(bitCloud).eq(0).And(
        qa.bitwiseAnd(bitCloudShadow).eq(0)).And(
        qa.bitwiseAnd(bitDilatedCloud).eq(0))

    opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2)
    thermalBand = image.select('ST_B6').multiply(0.00341802).add(149.0)
    return image.addBands(opticalBands,None,True)    \
                .addBands(thermalBand,None,True)    \
                .updateMask(mask)

def changBands(image):
    return
```

```
image.select(['SR_B1','SR_B2','SR_B3','SR_B4','SR_B5','SR_B7'],['SR_B2','SR_B3','SR_B4','SR
_B5','SR_B6','SR_B7'])

def add_VIs(image):
    NDVI=image.normalizedDifference(['SR_B5','SR_B4']).toDouble().rename('NDVI')
    EVI=image.expression(
        '2.5*((NIR-RED)/(NIR+6*RED-7.5*BLUE+1))',{
            'NIR':image.select('SR_B5'),
            'RED':image.select('SR_B4'),
            'BLUE':image.select('SR_B2')
        }).toDouble().rename('EVI')
    LSWI=image.normalizedDifference(['SR_B5','SR_B6']).toDouble().rename('LSWI')
    NDSVI=image.normalizedDifference(['SR_B6','SR_B4']).toDouble().rename('NDSVI')
    return image.addBands(NDVI).addBands(EVI).addBands(LSWI).addBands(NDSVI)


boundary='F:/hani /boundary.shp'
bound=geemap.shp_to_ee(boundary,encoding='gb2312')
train01='F:/hani/1999.shp'
trainP1999=geemap.shp_to_ee(train01,encoding='gb2312')
train02='F:/hani /2019.shp'
trainP2019=geemap.shp_to_ee(train02,encoding='gb2312')

lc5=ee.ImageCollection('LANDSAT/LT05/C02/T1_L2')  \
        .filterBounds(bound)  \
        .map(cloud_landsat5)  \
        .map(changBands)  \
        .map(add_VIs)

lc7=ee.ImageCollection("LANDSAT/LE07/C02/T1_L2")  \
        .filterBounds(bound)  \
        .map(cloud_landsat5)  \
        .map(changBands)  \
        .map(add_VIs)

lc8=ee.ImageCollection("LANDSAT/LC08/C02/T1_L2")  \
        .filterBounds(bound)  \
        .map(cloud_landsat8)  \
        .select(['SR_B2','SR_B3','SR_B4','SR_B5','SR_B6','SR_B7'])  \
        .map(add_VIs)

lc5_199800_ftp=lc5.filterDate('1998-01-01','2001-01-
01').filter(ee.Filter.dayOfYear(1,126)).select(['SR_B4','SR_B5','SR_B6','SR_B7','NDVI','EVI','LS
WI','NDSVI']).median()
```

```
lc5_199800_ghp=lc5.filterDate('1998-01-01','2001-01-
01').filter(ee.Filter.dayOfYear(126,300)).select(['SR_B4','SR_B5','SR_B6','SR_B7','NDVI','EVI','L
SWI','NDSVI']).median()
lc5_199800_bands=lc5_199800_ftp.addBands(lc5_199800_ghp)
lc5_199800=lc5_199800_bands.clip(bound)

image_Bands=lc5_199800.bandNames()
classcode='code1999'

samples_with_images=lc5_199800.sampleRegions(
    collection=trainP1999,
    properties=[classcode],
    geometries=True,
    scale=30
)
samples_with_images=samples_with_images.randomColumn('rand')
training=samples_with_images.filter(ee.Filter.lt('rand',0.7))
validation=samples_with_images.filter(ee.Filter.gte('rand',0.7))

training_with_samples=lc5_199800.sampleRegions(
    collection=training,
    properties=[classcode],
    geometries=True,
    scale=30
)

classifier=ee.Classifier.smileRandomForest(500).train(
    features=training_with_samples,
    classProperty=classcode,
    inputProperties=image_Bands
)

classified_199800=lc5_199800.classify(classifier=classifier,outputName='classified_199800')

validation_with_classified=classified_199800.sampleRegions(
    collection=validation,
    properties=[classcode],
    geometries=True,
    scale=30
)
testConfustionMatrix=validation_with_classified.errorMatrix(classcode,'classified_199800')
print('validation overall accuracy: ',testConfustionMatrix.accuracy().getInfo())
print('validation kappa accuracy: ',testConfustionMatrix.kappa().getInfo())
print('validation producersAccuracy accuracy:
```

```
',testConfustionMatrix.producersAccuracy().getInfo())
print('validation consumersAccuracy accuracy: ',testConfustionMatrix.consumersAccuracy().getInfo())

geemap.ee_export_image_to_drive(classified_199800,description='classified_199800', region=bound.geometry(), scale=30)


lc5_201011=lc5.filterDate('2010-01-01','2012-01-01')
lc7_2012=lc7.filterDate('2012-01-01','2013-01-01')
lc57=lc5_201011.merge(lc7_2012)
lc57_201012_ftp=lc57.filterDate('2010-01-01','2013-01-01').filter(ee.Filter.dayOfYear(1,126)).select(['SR_B4','SR_B5','SR_B6','SR_B7','NDVI','EVI','LSWI','NDSVI']).median()
lc57_201012_ghp=lc57.filterDate('2010-01-01','2013-01-01').filter(ee.Filter.dayOfYear(126,300)).select(['SR_B4','SR_B5','SR_B6','SR_B7','NDVI','EVI','LSWI','NDSVI']).median()
lc57_201012_bands=lc57_201012_ftp.addBands(lc57_201012_ghp)
lc57_201012=lc57_201012_bands.clip(bound)

classified_201012=lc57_201012.classify(classifier=classifier,outputName='classified_201012')
```